ÍNDICE

Resumen

El Comercio Electrónico o e-commerce representa una nueva forma de hacer negocios, brinda beneficios y oportunidades, y su perspectiva a futuro es muy prometedora. Este trabajo intenta enfocarse en la realidad del Ecuador y emprender en la materialización de un caso práctico llevándolo desde su planeación hasta su construcción e implementación. Por otro lado, se da fundamental relevancia a la implementación de mecanismos de seguridad de la información para garantizar integridad y confidencialidad de la misma.

Abstract

Electronic Commerce or e-commerce represents a new way of doing business, it gives benefits and opportunities, and its perspective is promising in the near future. This work tries to focus in the reality of Ecuador and undertake in working in a practical case taking it from planning to development and implementation. On the other hand, this work gives paramount importance to the implementation of information security mechanisms to guarantee integrity and confidentiality of it.

Universidad de Cuenca

Facultad e Ingeniería

Maestría en Telemática

"Desarrollo de una tienda virtual de Comercio Electrónico"

PROYECTO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN TELEMÁTICA

Alumno: Ing. Luis Roberto Jácome G.

Director: Ph.D. Diego Ponce Vásquez

29 de julio del 2010

Cuenca - Ecuador

El contenido de esta tesis es de absoluta responsabilidad del autor

Dedicatoria:

Este trabajo está dedicado a mi padre, mi madre, mi hermana, mi sobrina Mónica Estefanía y a toda mi familia quienes han confiado en mí incluso en los momentos difíciles

Capítulo 1

Introducción

1.1 Introducción teórica

1.1.1 El Comercio Electrónico

El Comercio Electrónico (e-commerce – electronic commerce) se refiere a realización de transacciones comerciales a través de medios tecnológicos como teléfono, fax, redes de computadoras, etc. Con la masificación del uso del Internet, se tiene previsto un crecimiento importante del Comercio Electrónico en un futuro cercano.

Países como Estados Unidos, Singapur, Japón llevan la delantera en cuanto a la expansión del Comercio Electrónico a diferencia de otros países en los que se tiene un uso más limitado del mismo.

Los factores que inciden en la falta de uso del Comercio Electrónico son muy variados y tienen que ver con situaciones económicas, infraestructura tecnológica, aspectos demográficos como edad, nivel de educación; factores culturales, etc., es relevante destacar que otro de los aspectos que influye es la desconfianza existente entre los usuarios de Internet de revelar los datos de tarjetas de crédito por Internet, este temor tiene su justificación debido a la gran cantidad de estafas que se producen cada año por parte de personas mal intencionadas que acechan en el Internet buscando precisamente estos datos confidenciales. Por ello, es fundamental manejar estos datos de la forma más cautelosa, si se quiere incursionar en el mundo del Comercio Electrónico, tanto por parte del cliente como del vendedor. El cliente solamente debe proporcionar sus datos de tarjetas de crédito en sitios seguros, por su parte el vendedor debe utilizar técnicas de seguridad en la transmisión de la información que permitan completar las transacciones de forma segura sin poner en evidencia la información del cliente.

Así mismo, la adopción del comercio electrónico en países en desarrollo es de vital importancia, principalmente desde el punto de vista del consumidor, puesto que se tiene la posibilidad de adquirir bienes que antes se los encontraba solamente en las grandes metrópolis. El comercio electrónico está reduciendo la brecha entre los países que tienen una gran variedad de productos y aquellos que tienen una limitada disponibilidad. Un propósito fundamental del comercio electrónico es proporcionar a

los consumidores la facilidad de adquirir productos que se encuentran en lugares distantes y que sería muy difícil de conseguir de la manera tradicional.

a) Antecedentes

a1) La brecha digital

Cabe citar el término “brecha digital”, se lo define como la diferencia en el uso y penetración de las TICs (Tecnologías de Información y Comunicaciones) entre países desarrollados y países en desarrollo. Cada día es más evidente su importancia en el desarrollo económico y cultural de la población, puesto que brinda ventajas a quienes gozan de su uso y priva de oportunidades a quienes sufren su carencia de uso.

En los países en los que la población tiene amplio acceso a las TICs, es más fácil impulsar proyectos de Comercio Electrónico, en cambio en países en los que el acceso a las TICs es más limitado, es más difícil implantar exitosamente proyectos de Comercio Electrónico, este aspecto se puede explicar debido al hecho de que mientras más se utiliza herramientas tecnológicas como el Internet, más dominio se tiene del mismo y se llega a obtener más confianza en su utilización, hasta el punto de llegar a crearse nuevas expectativas, incluyendo la posibilidad de comprar productos por Internet sin necesidad de moverse de casa o del trabajo.

a2) El Comercio Electrónico dentro del e-Business

Podemos destacar los siguientes tipos de e-business: B2B – Business to Business, B2C - Business to Consumer y C2C – Customer to Customer.

Para el tipo B2B, se trata de transacciones hechas entre empresas, se destacan principalmente por ser el tipo de Comercio Electrónico que más volumen de dinero mueve a nivel mundial. Para el tipo B2C, se caracterizan por ser transacciones entre empresas y consumidores finales, ejemplos de este tipo son las tiendas y negocios virtuales. Por otra parte, tenemos el C2C que se refiere al intercambio de productos entre consumidores finales, ejemplos de este tipo son los sitios de subastas en línea.

En lo referente al sector público, se tiene el tipo G2G al hacer referencia a transacciones entre entidades públicas, G2B es el caso de negocios entre empresas públicas y privadas, por ejemplo, las compras públicas y finalmente G2C cuando la empresa pública llega al ciudadano. Se debe recalcar que cuando las entidades públicas intervienen, esta clasificación es referida como e-Government.

	Gobierno	Empresas	Personas
Gobierno	G2G Coordinación y transferencia de información	G2B Información y servicios	G2C Información y servicios
Empresas	B2G Trámites, impuestos, información	B2B Comercio electrónico	B2C Comercio electrónico
Personas	C2G Impuesto y trámites	C2B Laborales	C2C Compra/Venta Remates

Tabla 1.1 Tipos de e-business

Así mismo, cabe señalar la clasificación de las empresas de acuerdo a la actividad que realizan en el Internet, tenemos primeramente las empresas “Brick and Mortar”, que son aquellas que realizan sus actividades complemente de manera tradicional y no tiene presencia en Internet. En segundo lugar, tenemos las empresa “Click and Mortar”, que son aquellas que tienen como canal principal de ventas su negocio de manera tradicional, pero también realizan ventas a través del Internet. Y Finalmente tenemos las “empresas enteramente digitales” que realizan sus actividades comerciales completamente por Internet. Por su parte a las empresas que venden artículos al menudeo por Internet se los conoce como e-tailers o e-detallistas.

Por su parte, si tomamos en cuenta la clasificación de los sitios de Comercio Electrónico podemos encontrar los siguientes:

- **Informacional:** solamente sirve para mostrar contenido a los visitantes
- **Sitio web interactivo:** intercambio de información en 2 sentidos
- **Atractores:** atrae e interactúa con visitantes. Ej: juegos on-line.
- **Transaccional:** permite procesar transacciones completas a través de Internet.

Al hablar de empresas que realizan ventas a través de Internet, se debe clasificar sus productos en 2 categorías: productos físicos y productos digitales. Para la venta

de productos físicos como libros, relojes, computadoras, etc., obviamente se necesita tener en consideración los costos que representa el transporte de los mismos hacia el cliente. Por su parte, para los productos digitales el transporte no es un aspecto importante puesto que su distribución se la realiza a través del mismo Internet. Como ejemplos de productos digitales tenemos software, música, suscripciones, membrecías, servicios, consultorías, etc.

a3) El gran reto: conseguir visitantes

Uno de los aspectos relevantes después de la puesta en marcha de un sitio web tiene que ver con conseguir visitantes al mismo, y en el caso de una tienda virtual, se debe tratar de que estos visitantes se conviertan en compradores, para garantizar la rentabilidad del emprendimiento.

Se tiene muchas técnicas para generar tráfico hacia el sitio web, de las que podemos destacar el marketing por Internet, que tiene que ver con anunciar el sitio de diversas formas, por ejemplo el intercambio de enlaces, buscar el buen posicionamiento en los principales motores de búsqueda como Google mediante el uso de SEO (Search Engine Optimization), la utilización de la web 2.0 con las redes sociales, la utilización de programas de publicidad como Google adWords, etc.

a4) El ocaso de los punto com

Después del año 2000 se creó gran expectativa en países como los Estados Unidos sobre las posibilidades del Comercio Electrónico y por ello muchas de empresas incursionaron en desarrollo de negocios virtuales, sin embargo las expectativas creadas en ese entonces, no se cumplieron y muchos de los emprendimientos fracasaron y las tiendas virtuales desaparecieron. Es muy importante tomar en cuenta esta experiencia y tener conciencia de que la puesta en marcha de un negocio virtual no garantiza su éxito ni su aceptación por parte de los usuarios y por tanto su rentabilidad. Sin embargo, al aprender de esta experiencia se tiene un nuevo enfoque en el que el Comercio Electrónico es viable siempre y cuando se tomen en consideración aspectos que permitan el éxito de las tiendas virtuales, estos aspectos fundamentales son: pensar en la rentabilidad, un plan de negocios, actualización constante de los contenidos de la tienda virtual, un diseño gráfico agradable, uso de técnicas de seguridad para garantizar confidencialidad de las transacciones, etc.

Se cree en el enorme potencial del Comercio Electrónico y se estima un crecimiento vertiginoso en los próximos años, sin embargo se debe tener presente los riesgos existentes tanto para el cliente como para el vendedor.

a5) El Comercio Electrónico y la Telemática

Se considera la Telemática como una ciencia en la que se fusionan las Telecomunicaciones y la Informática, puesto que se utilizan medios de comunicación como las redes de ordenadores y las redes telefónicas para realizar diferentes tipos de procesamientos y transacciones asistidos por computadoras.

El Comercio Electrónico es considerado parte de la Telemática puesto que trata el procesamiento de transacciones comerciales seguras a través de redes abiertas e inseguras como el Internet.

a6) Innovaciones tecnológicas

En lo referente a las innovaciones tecnológicas a implementar en el presente proyecto podemos citar el uso de la arquitectura de 3 capas mediante el modelo MVC (Modelo-Vista-Controlador), el uso de la tecnología AJAX (Asynchronous JavaScript and XML) en el lado del cliente para mejorar los tiempos de respuesta de la tienda virtual, la utilización de Web Services y SOAP para permitir la comunicación con otras aplicaciones en Internet y la utilización del Framework JCA (Java Cryptography Architecture) y OpenSSL para implementar los mecanismos de seguridad de la información.

b) Estado del arte

b1) Realidad mundial

El comercio electrónico está en pleno desarrollo a nivel mundial, países como Singapur, Japón, Malasia y Estados Unidos experimentan una masificación del Comercio Electrónico y se ha generado el término de “Economía Digital” para hacer referencia al volumen de transacciones hechas a través de Internet y que influyen en la economía de estos países.

Sin embargo, el Comercio Electrónico aún tiene varios aspectos que resolver, como son la seguridad de las transacciones, mejoramiento de la logística de distribución de productos, leyes de protección a los consumidores, etc.

En América Latina, los países líderes en cuestión de Comercio Electrónico son Brasil, Argentina y México, en los que el volumen de transacciones hechas por Internet es considerable con respecto al total de transacciones hechas de manera tradicional. Podemos apreciar claramente que en nuestra región existen muchas oportunidades para el desarrollo del Comercio Electrónico.

b2) Situación en el Ecuador

En el Ecuador la situación del Comercio Electrónico está en sus inicios, si bien existen iniciativas de Comercio Electrónico, estas no son suficientes, puesto que falta por cubrir una gran cantidad de categorías de productos y servicios que se pueden ofrecer a través de Internet.

Existe una serie de obstáculos que se deben vencer para desarrollar el Comercio Electrónico en el Ecuador, por ejemplo la falta de infraestructura, puesto que un porcentaje pequeño de la población tiene acceso al Internet, cuestiones de seguridad debido a la gran cantidad de estafas que se registran por el uso indebido de tarjetas de crédito, la desconfianza por parte de los consumidores a los negocios en Internet, la falta de regulación del Comercio Electrónico en el país, factores culturales, falta de logística adecuada en la entrega de productos, dificultad para realizar reclamos y devolución de productos, etc.

Si bien, existen obstáculos que sortear, también es importante recalcar que las oportunidades son grandes debido al continuo desarrollo de la tecnología, la disminución de los precios de los equipos, y el incremento del número de usuarios de Internet.

b3) El negocio tradicional

Se ha considerado el caso del almacén J. & G. Exclusividades, que es un negocio de venta de adornos para el hogar, artículos religiosos (crucifijos, ángeles, imágenes), artículos de colección. Realiza sus ventas al menudeo de manera tradicional (empresa detallista Brick and Mortar), tiene su sede en la ciudad de Loja y ha operado por un periodo de 10 años. Ha consolidado su presencia en esta ciudad, sin embargo se tiene la expectativa de dar a conocer el negocio en otras ciudades del Ecuador. Así mismo, se tiene planificado establecer su presencia en el Internet con el afán de promocionar el almacén y la expectativa de abrir un nuevo canal de ventas que permita abarcar un mayor número de clientes y obtener mejores oportunidades de negocio. (convertirse en una empresa e-detallista Click and Mortar).

1.2 Descripción del problema y/o necesidad

a) Problemas a ser resueltos

Existe muy poca difusión y desarrollo del Comercio Electrónico en la ciudad de Loja, existen pocos emprendimientos en lo que se refiere a negocios B2C Business to Customer, y negocios detallistas (ventas al menudeo), las causas principales tienen que ver con la falta de regulación que proteja a los consumidores, asuntos culturales, asuntos económicos puesto que sólo un porcentaje de la población utiliza tarjetas de crédito, poco desarrollo de la tecnología, solo un porcentaje pequeño de la población tiene acceso a Internet y sólo el 1% tiene Internet de banda ancha[1], existe falta de confianza en los negocios electrónicos.

Por su parte, el almacén J. & G. Exclusividades opera solamente en la ciudad de Loja y por tanto no tiene presencia en otras ciudades del Ecuador, así mismo, no ha utilizado medios tecnológicos para la promoción del negocio.

Otro problema del negocio en estudio es la imposibilidad de consultar sobre las existencias de productos desde cualquier lugar utilizando el Internet.

Así mismo, podemos citar la pérdida de oportunidades de negocio por la falta de presencia online, lo que conlleva a la necesidad de incursionar en el Comercio Electrónico puesto que los negocios deben evolucionar y emprender en esta área si quieren tener una medio que permita alcanzar un mercado global con miles de nuevos clientes potenciales que no se consiguen trabajando de la manera tradicional.

Para el desarrollo del proyecto se deben abordar problemas técnicos relacionados a las aplicaciones web, como es la construcción de un sistema de 3 capas: Capa UI (User Interface), Capa de Negocio y Capa de base de datos, además se debe considerar temas como el mantenimiento de la tienda virtual, administración de la base de datos, etc.

Existen además, aspectos que no son técnicos y que deben ser considerados para el cumplimiento del proyecto, como son la publicación y actualización de los productos, la administración y manejo de las órdenes de compra, la promoción de la tienda virtual. Se debe resaltar que los aspectos no técnicos del proyecto serán solventados por los dueños del negocio.

[1] (Supertel 2009), (CONATEL 2010)

b) Necesidades a ser satisfechas

Las pocas iniciativas de comercio electrónico existentes en la ciudad de Loja, que proporcionen confianza a los consumidores y sean eficientes en sus operaciones.

La escasez de alternativas que tienen los consumidores en la ciudad de Loja en cuanto se refiere a la compra de productos y servicios por Internet y su respectiva entrega en sus domicilios.

La carencia de oportunidades de negocios por no tener presencia en Internet.

Resulta evidente la necesidad de construir un sitio web dinámico para el almacén que permita abordar los problemas existentes, dando principal importancia a la creación de un nuevo canal de ventas que permita al almacén alcanzar nuevos mercados, así mismo es de fundamental importancia el requerimiento de una fácil administración del negocio y control de existencias de productos utilizando este nuevo sitio web.

1.3 Justificación del proyecto de tesis

1.3.1 Justificación técnica

El comercio electrónico es considerado como parte de la Telemática, puesto que permite realizar transacciones a través de la red de redes y su operación es más compleja que las transacciones llevadas a cabo de manera tradicional.

En este proyecto existe un importante componente de investigación y desarrollo, puesto que se debe construir una solución que esté a la par de las últimas características y tendencias tecnológicas existentes. Es relevante esta consideración debido al continuo adelanto e innovación de las tecnologías.

El desarrollo de una tienda virtual de comercio electrónico es un tema complejo, debido a que se requiere de conocimiento en cuanto al desarrollo de aplicaciones para la web, puesto que el maestrante necesita considerar aspectos como la instalación, configuración y administración de programas especializados como bases de datos, servidores web; el diseño de algoritmos, la codificación de la aplicación en un lenguaje de programación orientado al web, el diseño gráfico de la interfaz del sitio, el mismo que tiene que ser agradable y fácil de utilizar, etc.

Existen temas críticos como la seguridad de las transacciones, cuya consideración implica el dominio de temas como criptografía, respaldo de bases de datos, etc.

En cuanto a innovación tecnológica se ha planeado utilizar tecnologías como Servicios Web que facilitan la comunicación con otras aplicaciones en Internet sin importar la plataforma ni el lenguaje de programación utilizado, dan la posibilidad de reutilización de los datos y permiten una efectiva distribución de la aplicación. En cuanto al desempeño de la tienda virtual, se tiene pensado optimizar el tiempo de respuesta por medio de AJAX, que permite la descarga de contenido sin afectar la visualización del cliente. Finalmente, se ha optado por la arquitectura de 3 capas implementada por el modelo MVC (Modelo – Vista – Controlador) de J2EE de Java, es importante recalcar que la tecnología J2EE tiene componentes como los Servlets que permiten interactuar incluso con dispositivos celulares que poseen navegación por Internet.

1.3.2 Justificación operativa y económica

Para el éxito de la implantación de una tienda virtual es importante la consideración de aspectos que no son técnicos por ejemplo el ingreso y actualización de productos, el manejo de pedidos, la promoción de la tienda virtual, la planificación y definición de estrategias de negocio, etc. Todos estos aspectos mencionados junto con los gastos que representan van a ser afrontados por los dueños del negocio, por ser la parte beneficiada del proyecto, puesto que se cree necesaria la presencia de la empresa en el Internet y de los beneficios que pueda traer.

1.3.3 Justificación legal y/o regulatoria

En lo referente a la justificación legal, se tiene que el negocio escogido para la implementación del proyecto, está constituido legalmente y ha operado desde hace 10 años, cumpliendo con las normas establecidas por organismos de control como el SRI.

1.3.4 Beneficios para la comunidad o institución a quien está dirigida la tesis.

Este proyecto pretende ser uno de los pioneros en la ciudad de Loja, en lo referente al Comercio Electrónico. Se quiere brindar a la comunidad facilidad y comodidad en la compra de artículos, puesto que el cliente no necesita desplazarse hacia la tienda física, lo puede realizar a cualquier hora, sus productos son entregados en su domicilio y se tendrá la confianza de que la compra es segura y confiable.

Los beneficios para la pequeña empresa a la que está dirigida este desarrollo, consisten en abrir un nuevo canal de ventas, como es el Internet, con lo que se pretende ampliar considerablemente el número de potenciales clientes, no solo a nivel local sino regional y en el exterior. Así mismo es importante recalcar que con este emprendimiento el negocio tradicional se verá beneficiado por la publicidad que

se obtiene al contar con su versión en Internet, por lo que las oportunidades de negocio se verán multiplicadas. Existen beneficios intangibles como el prestigio y mejoramiento de la imagen del negocio que se obtiene al utilizar técnicas modernas de promoción y ventas como son las ventas por Internet; obtener enormes ventajas sobre la competencia, conseguir nuevos clientes y afianzar los ya obtenidos al ofrecerles un nuevo canal de comunicación; estar a la par de los últimos adelantos tecnológicos en cuestión de Comercio Electrónico.

1.4 Objetivos.

1.4.1 Objetivo general

- Desarrollar una tienda virtual de comercio electrónico para el almacén J. & G. Exclusividades.

1.4.2 Objetivos específicos

- Establecer el plan de negocios para Internet del almacén J. & G. Exclusividades.

- Desarrollar una aplicación web utilizando una arquitectura three tier que permita una separación clara entre la capa de base de datos, la capa de negocios y la de presentación o UI.

- Implementar técnicas de gestión de usuarios que permitan la distinción de roles de usuarios, de modo que nos ayude a la administración de la aplicación y al fácil acceso y seguridad de usuarios/clientes del negocio.

- Determinar e implementar las formas de pago más adecuadas para la tienda virtual.

- Desarrollar Web Services que permitan el acceso a los datos de la base y demás métodos que contribuyan a mejorar la seguridad, eficiencia y reutilización de código en nuestra aplicación.

- Desarrollar una solución del tipo transaccional que permita completar ventas por Internet.

- Establecer las políticas de seguridad de la información que servirán de guía para la administración de la tienda virtual.

- Determinar e implementar las formas de promoción y marketing por Internet más eficientes para la tienda virtual.

1.5 Alcance del proyecto

El proyecto consiste en el desarrollo de una tienda virtual de comercio electrónico, de tipo B2C (Business to Customer) transaccional e-detallista, es decir una tienda de venta virtual al menudeo, para el almacén J. & G. Exclusividades de la ciudad de Loja. Las características más relevantes del proyecto son las siguientes:

- Se necesita desarrollar una estrategia de comercio electrónico, se debe analizar las fortalezas y debilidades del negocio, así como las oportunidades y amenazas del mercado, para establecer un plan de negocios que viabilice el éxito del emprendimiento. Es muy necesario considerar temas sociales como el impacto de la tienda virtual, debido a que si no presenta una solución efectiva a los consumidores, este proyecto podría no tener la aceptación y el uso esperado.

- Se requiere desarrollar la capa de interfaz de usuario (interfaz gráfica de la tienda virtual), cuya apariencia sea agradable y esté conforme a la imagen corporativa del negocio (utilizar logotipos, colores, slogans de la empresa, información de la empresa, página para contactos, etc.). Se deberá incluir una sección con las FAQs (Frecuently Asked Questions) o preguntas más frecuentes para que los clientes puedan entender de forma clara y concisa el modo de operación de la tienda virtual.

- Se tiene que desarrollar las capas de negocio y de base de datos de la tienda virtual, que forman la parte que soporta al sitio web, se debe considerar el análisis de requerimientos, el diseño lógico, el diseño físico, la instalación y configuración de la base de datos, la instalación y configuración del software necesario para la implementación del sitio.

- Se desarrollará un módulo para la fácil administración de la tienda virtual, en el que un digitador (sin necesidad de poseer conocimientos técnicos) podrá ingresar, editar, actualizar los datos de los productos, subir fotografías y videos. Por su parte el vendedor encargado podrá revisar, atender y hacer seguimiento de pedidos y órdenes de compra, consultar reportes de ventas, revisar y responder consultas de clientes, editar categorías de productos, etc. Por tanto, la administración de la tienda virtual se la hará completamente vía web.

- Se implementará un módulo de inventario de los productos de la tienda virtual, el mismo que constituye la base para la operación de la tienda virtual.

- Se deberán redactar las políticas de seguridad de la información que servirán como guía para la correcta administración y operación de la tienda virtual con el objetivo de solventar este aspecto crítico y prevenir posibles ataques informáticos tanto externos como internos al negocio.

- Una de las expectativas creadas en este proyecto es la utilización de herramientas de código abierto (open source), para el desarrollo e implementación de la tienda virtual, se ha creído conveniente esta iniciativa debido a las altas prestaciones que brinda las herramientas de software libre, especialmente del lado del servidor y herramientas para desarrolladores (sistemas operativos: Linux, bases de datos: MySQL, servidores web: Apache, lenguajes de programación: J2EE (JSP, Servlets, JavaBeans), JavaScript en el lado del cliente; herramientas de diseño: Gimp, modelado de datos: ArgoUML, Dia, Umbrello, etc.), dando preferencia sobre las herramientas de pago propietarias de Microsoft y otras marcas. Así mismo se debe destacar las ventajas de las licencias GPL (General Public License), en las que el desarrollador puede utilizar estas herramientas sin ningún costo.

- Se incorporará el uso de Web Services (Servicios Web) y de protocolos SOAP (Simple Object Access Control) para la búsqueda avanzada de productos.

- Se utilizará AJAX (Asynchronous JavaScript and XML) en el lado del cliente para mejorar el rendimiento de la aplicación y disminuir el tiempo de carga de la información de los productos de la tienda virtual.

- Es pertinente tener en cuenta el desarrollo de contenidos, que tiene que ver con la publicación de los productos a comercializar, edición de fotografías, promociones, actualización de productos, etc. Esta tarea será realizada por el digitador/operador asignado por el dueño del negocio, sin embargo es pertinente la asesoría y supervisión del desarrollador para este tipo de tareas.

- Se dispondrá de una versión de la tienda virtual en inglés con el fin de alcanzar mercados en otros países y promocionar el contenido para audiencias diferentes utilizando el inglés por ser el lenguaje de mayor aceptación comercial.

MARCO TEÓRICO

Capítulo 2

PLAN DE NEGOCIOS

2.1 Introducción teórica

2.1.1 e-business

El e-business[2] se refiere a una visión más amplia que el e-commerce (comprar y vender bienes y servicios), se refiere además a la utilización de las TICs (Tecnologías de la Información y Comunicaciones) para soportar el comercio y mejorar el rendimiento del negocio. Algunos de los aspectos que caracterizan al e-business son los siguientes:

- Dar servicio a los clientes.
- Colaboración con los socios de negocios.
- Utilizar técnicas de e-learning.

Por su parte el e-business tiene que ver con la toma de decisiones de una empresa. Es fundamental el uso de las TICs con el fin de que el experto del negocio pueda realizar más rápidas y mejores decisiones.

Dentro del e-business debemos definir algunos conceptos importantes:

2.1.1.1 Modelo de negocios

Método de hacer negocios, por medio del cual una empresa puede generar ingresos para su auto sostenimiento. El modelo explica la cadena de valor de una empresa. Los modelos de negocios son un subconjunto de un plan de negocios o un caso de negocio.

2.1.1.2 Cadena de valor

Son las actividades que añaden valor para que una empresa alcance sus objetivos en las diferentes etapas del proceso de producción.

[2] (González, Introducción al E-Business 2009)

2.1.1.3 Plan de negocios

Documento que evidencia los objetivos del negocio y define los pasos para alcanzarlos.

2.1.1.4 Caso de negocio

Es un documento utilizado por la gerencia de una empresa para guardar fondos para un proyecto específico, dando principal énfasis a la justificación de una inversión específica.

Todos los modelos de negocios deben dar a conocer el modelo de ingresos y su propuesta de valor. Los modelos de ingresos se refieren a las actividades que realizará el negocio para obtener ingresos económicos, por otra parte la propuesta de valor se refiere a los beneficios no cuantificables que la empresa obtiene por el uso del Comercio Electrónico.

2.1.1.5 Business Intelligence

Se refiere a los procesos, tecnologías y herramientas requeridas para transformar datos en información, información en conocimiento y conocimientos en planes que orienten a la empresa. Business Intelligence tiene que ver con los data warehouses, herramientas analíticas de negocios y gestión del conocimiento.

2.1.1.6 Data warehouse

Un data warehouse consiste en un repositorio de datos históricos y detallados y que está disponible a los usuarios de modo que facilite la acertada toma de decisiones.

2.1.1.7 Redes sociales

Las redes sociales son sitios web que comunican a personas con intereses similares, muchas veces brindándoles el servicio de forma gratuita. Ejemplos de servicios ofrecidos por las redes sociales son: intercambio de fotos, videos, textos, creación de comunidades, blogging, etc. El comercio electrónico puede beneficiarse de las redes sociales debido a que permite alcanzar a una amplia audiencia y aprovechar la interacción de los miembros de las redes sociales para llegar a un mayor número de personas.

2.1.1.8 Mercadeo one to one

Es una forma de hacer mercadeo que trata a cada cliente en forma individual para enfocar la publicidad a las necesidades del cliente, es decir, se proyecta a la

personalización de los bienes y servicios que ofrece la empresa. Para llegar a realizar este tipo de mercadeo, es evidente que la empresa debe conocer el perfil del cliente (preferencias, comportamientos, rasgos demográficos). Para construir una base de datos con los perfiles del cliente es necesario pedir esta información al cliente mediante formularios en el sitio web o utilizar otras técnicas como encuestas, estudios de mercado, etc.

2.1.1.9 CRM

El CRM (Customer Relationship Management)[3] se basa en un enfoque de servicio al cliente en el que se proyecta una relación duradera con el cliente que provea beneficios al mismo y valor agregado a la organización. Existen técnicas de CRM como "cross-sell" en la que se ofrecen productos complementarios o "up-sell" en la que se realzan otros productos que han tenido éxito con otros clientes. Otras técnicas de CRM incluyen la segmentación de clientes y la personalización de productos y servicios.

El CRM brinda beneficios como poder escoger fácilmente productos y servicios, permite una resolución rápida de problemas y ofrece fácil y rápido acceso a la información, además se debe resaltar beneficios para el cliente como el soporte post-compra de productos.

2.2 Desarrollo del Plan de Negocios de la tienda virtual.

Para la implementación de la tienda virtual de Comercio Electrónico se ha considerado el caso del almacén J. & G. Exclusividades, que es un negocio de venta de adornos para el hogar, artículos religiosos (crucifijos, ángeles, imágenes), artículos de colección. Realiza sus ventas al menudeo de manera tradicional (empresa detallista Brick and Mortar), tiene su sede en la ciudad de Loja y ha operado por un periodo de 10 años. Ha consolidado su presencia en esta ciudad, sin embargo se tiene la expectativa de dar a conocer el negocio en otras ciudades del Ecuador. Así mismo, se tiene planificado establecer su presencia en el Internet con el afán de promocionar el almacén y la expectativa de abrir un nuevo canal de ventas que permita abarcar un mayor número de clientes y obtener mejores oportunidades de negocio (convertirse en una empresa e-tailer Click and Mortar).

[3] (González, Comportamiento del consumidor y CRM 2009)

2.2.1 Análisis de mercado

Los productos ofrecidos en el almacén J. & G. exclusividades varían desde adornos para el hogar (figuras de cerámica fina – de resina – de alabastro, cuadros, lámparas decorativas); artículos religiosos (crucifijos, imágenes religiosas, figuras de ángeles, repisas talladas en madera); artículos de colección (autos, motos, aviones de colección a escala, radios, teléfonos con estilos clásicos, relojes de pared); artículos de marcas reconocidas (encendedores Zippo, navajas suizas Victorinox, relojes Casio, Fossil). Para los negocios tradicionales que quieren abrir un nuevo canal de ventas en Internet, no necesariamente deben incluir todos los productos existentes en el negocio tradicional, más bien se deben identificar los productos que pudieran tener mayor aceptación, sean más fáciles de transportar, su precio sea razonable, etc.

Existen temporadas en el año en los que los productos que exhibe el negocio son muy solicitados, por ejemplo: Navidad, San Valentín, Día del Padre, Día de la Madre, etc., por tanto en fechas como estas, es relevante destacar estos productos debido a la alta demanda. Sería muy conveniente para los clientes tener la posibilidad de escoger este tipo de productos y comprarlos desde casa o desde el lugar del trabajo, y que los productos sean entregados en su mismo hogar.

Por otra parte, es una buena idea vender artículos de marcas reconocidas puesto que el cliente reconoce dicha marca y conoce las características de estos productos, por ello está seguro de lo que compra y la calidad que espera obtener.

Los artículos de colección son bien vendidos en Internet, muchas veces no están disponibles en tiendas tradicionales y coleccionistas no tienen reparo en adquirirlos.

Los artículos religiosos (artesanías talladas en madera) como crucifijos e imágenes son bien aceptadas principalmente en el extranjero, por lo que pueden ser consideradas para su promoción. Es imprescindible la protección adecuada de estos productos en el transporte de los mismos para evitar reclamos por rayones, partes rotas, etc.

En cuanto a la selección de los productos por el precio, cabe recalcar que solamente es factible ofrecer artículos con un precio superior a los 10 dólares, puesto que al utilizar pagos con tarjeta de crédito, el banco emisor de la tarjeta cobra una comisión por sus servicios, por lo tanto, es importante esta consideración para garantizar la rentabilidad de la transacción. Los micropagos (montos inferiores a 10 dólares) no son considerados en este proyecto.

Es importante hallar el nicho de mercado para el que está dirigida la tienda virtual y plantear estrategias orientadas hacia ellos[4]. Por una parte debemos considerar que solo un porcentaje de la población tiene acceso al Internet, así mismo si tomamos en cuenta las características de los productos ofrecidos en el almacén, podemos afirmar que el público objetivo de la tienda virtual serían personas de la clase media y alta. Por otra parte, estudios demuestran que existen grupos de personas que estarían dispuestos a comprar por Internet, incluso con mayor aceptación que con los negocios de manera tradicional. Estos grupos son:

- Usuarios con necesidad (discapacitados, ancianos)
- Jóvenes que sienten fascinación por el uso de las Tecnologías de la Información y Comunicaciones
- Estudiantes universitarios
- Profesionales jóvenes y profesionales con alto dominio del uso de Internet.
- Personas con acceso a tiempo completo a Internet (oficinistas, gerentes, empresarios, etc.).
- Consumidores muy ocupados que no pueden abandonar sus sitios de trabajo.
- Clientes del almacén que han encontrado útil la versión virtual del negocio y consideran ventajoso utilizar este medio para realizar los pedidos.

Podemos destacar que los factores que influyen en la aceptación del Comercio Electrónico consisten en: la situación económica, la edad (los jóvenes son los más entusiastas), el nivel de educación, disponibilidad y velocidad de acceso a Internet, etc.

2.2.2 Análisis FODA

2.2.2.1 Fortalezas

- El negocio tradicional cuenta con personal encargado de realizar inventarios y por tanto se puede utilizar este recurso humano para ingresar, modificar, actualizar los productos a la tienda virtual.

[4] (2createawebsite.com 2010)

- Se tiene el respaldo del negocio tradicional, por lo que se cuenta con el financiamiento para cubrir con los gastos de inversión y de operación. Por otra parte, se cuenta con una considerable cantidad y variedad de productos que se pueden publicar organizados en diferentes categorías.

- Se tiene amplia experiencia en negocios desarrollados en forma tradicional y se pueden aplicar muchas de las técnicas que han dado buenos resultados.

- Se puede utilizar las PCs del almacén para que los operadores/digitadores ingresen los datos, así mismo se puede utilizar las PCs para la administración de la tienda virtual. Se debe recalcar que es importante la protección mediante contraseñas para evitar ataques internos.

- Si bien los productos que ofrece el almacén J. & G. Exclusividades son productos tangibles, éstos no son perecibles y por tanto es factible su transporte sin preocuparse de fechas de expiración o caducidad como es el caso de los alimentos.

2.2.2.2 Oportunidades

Existen pocos emprendimiento de Comercio Electrónico en el Ecuador y mucho menos en la ciudad de Loja, por lo que existen una serie de oportunidades que se pueden aprovechar:

- Establecer un nuevo canal de comunicación con el cliente, en el que se puede agilitar pedidos, reclamos, atención al cliente, consultas, etc. Así mismo, brindar mayor comodidad a los clientes del negocio, mejorar la comunicación y la relación con los mismos, y por otra parte obtener nuevos clientes.

- Tener la posibilidad de convertirse en una empresa pionera de Comercio Electrónico en Loja y obtener el reconocimiento respectivo.

- Obtener grandes ventajas sobre competidores. Tener presencia en Internet permite alcanzar mercados que la competencia tradicional no tiene acceso, así mismo se obtiene una mejor imagen empresarial con respecto a la competencia ante los clientes que forman el mercado en disputa.

- Ganar posicionamiento en el mercado, ganar prestigio al utilizar herramientas tecnológicas modernas en comparación con los negocios que llevan a cabo sus actividades de forma tradicional.

- Incrementar el número de sitios web ecuatorianos y lojanos, se contribuye a crear contenido generado en Ecuador, los internautas ecuatorianos y lojanos

tendrán un sitio más con el que se puedan identificar. Esto es resaltable si consideramos los innumerables sitios existentes provenientes de otros países.

- Expansión del negocio, alcanzar mercados nacionales e internacionales, en donde el comercio electrónico es bastante aceptado.
- El almacén tiene productos hechos en el Ecuador (la mayoría de los artículos religiosos). Se tiene la oportunidad de dar a conocer estos productos en el extranjero y crear interesantes oportunidades de negocio. Se tiene beneficios tanto para el almacén como para los maestros artesanos.
- Disminución de costos por publicidad, utilizar el Internet en lugar de medios tradicionales como radio, prensa, televisión que resultan más costosos y en muchos casos son menos efectivos.
- Aprovechar las facilidades como imágenes, videos para promocionar productos.
- Dar a conocer promociones, ofertas de manera efectiva e inmediata.
- Mejor atención al cliente: la tienda virtual permite crear un poderoso canal 24/7 para el servicio y la información. La tienda virtual sirve para el crecimiento del negocio tradicional y aumentar la base de clientes.
- Tener un modelo de negocios multicanal: varios canales de distribución y ventas. Negocio tradicional, tienda virtual.
- Obtener experiencia (Know-How) en los negocios electrónicos, lo que puede representar una enorme ventaja competitiva a mediano y largo plazo.
- Crear un nuevo canal para retroalimentación y corregir errores de operación, servicio, calidad de productos, reclamos, etc.
- Tener la oportunidad de romper las viejas reglas y contribuir a la implantación de un nuevo modelo de compras para la comunidad.

2.2.2.3 Debilidades

- Empresa nueva en el Internet, por lo que no es conocida en el ciberespacio.
- Al ser una empresa pequeña y joven, se cuenta con un capital limitado.
- Falta de experiencia en negocios por Internet.

- El almacén no ofrece productos digitales, por lo que será más difícil enviar productos físicos.

2.2.2.4 Amenazas

- Potenciales pérdidas por estafas que se registran por el uso indebido de tarjetas de crédito.
- Desconfianza de los consumidores a los negocios en Internet.
- Falta de regulación del Comercio Electrónico
- Falta de infraestructura: un porcentaje pequeño de la población tiene acceso a Internet
- Falta de uso de compras debido a factores culturales de la población.

2.2.3 Plan de mitigación de riesgos

2.2.3.1 Análisis de amenazas

Todo emprendimiento (en Internet o tradicional) tiene sus riesgos que solventar, por su parte los negocios en Internet afrontan nuevos retos que deben ser considerados cuidadosamente para su implementación. Podemos abordar las siguientes amenazas del proyecto:

A: Bajo volumen de ventas, baja rentabilidad.

En caso de un bajo volumen de ventas, se puede tomar medidas para contrarrestar esta amenaza como el despliegue de mayor publicidad, la utilización de promociones atractivas para el cliente, por ejemplo descuentos si se realiza el pedido en la tienda virtual. Otra medida más extrema, sería disminuir los márgenes de utilidad, para que las ventas no disminuyan y poder solventar la empresa en tiempos de crisis.

A: Ataques de seguridad a la tienda virtual

Ingresos no autorizados a la base de datos.

Se puede contrarrestar esta situación, con políticas como cambio periódico de contraseña, utilización de contraseñas fuertes, respaldo diario de la base de datos (utilizar cintas magnéticas de gran capacidad), es recomendable que los datos confidenciales de los clientes sean encriptados en la base de datos.

Creación de órdenes ficticias.

Revisar el perfil del cliente y obtener información clave como teléfono, email para confirmar el pedido mediante estos medios.

Ataques de escucha de la información.

Para el despliegue de formularios en los que se piden los datos del cliente, utilizar el protocolo TLS, para establecer una comunicación segura.

Estafas que se registran por el uso indebido de tarjetas de crédito

Mediante la implantación de un servidor seguro se obtiene que las comunicaciones entre el cliente y la tienda virtual sean seguras, por tanto los datos financieros del cliente están fuera de peligro.

Ataques de negación de servicio.

Una medida para contrarrestar este ataque sería el despliegue de un servicio redundante, para que en caso de fallo o ataque DoS (Denial of Service – Negación de Servicio) se pueda recuperar inmediatamente las operaciones del portal.

A: Pocas visitas a la tienda virtual

Existen varias recomendaciones para generar tráfico hacia la tienda virtual y hacer que estos visitantes se conviertan en clientes.

- ✓ Publicar enormes cantidades de contenidos, mantener listados de productos renovados constantemente.
- ✓ Optimizar cada página para los motores de búsqueda. (SEO – Search Engine Optimization)
- ✓ Internet Marketing de la tienda virtual. Marketing a través de redes sociales: crear perfiles de la tienda virtual en sitios como Facebook, Twitter, etc., y captar seguidores para alentarlos a comprar en la tienda virtual. Predicciones en Internet dicen que estar bien posicionado en las redes sociales será a corto plazo tan importante como estar bien posicionado en Google.
- ✓ La construcción de la tienda virtual de Comercio Electrónico tiene 2 partes: una parte técnica y una parte administrativa. Luego de resolver los aspectos técnicos se requiere de mucho trabajo en lo referente a la publicación de contenidos de calidad.

- ✓ Unirse a foros relacionados a coleccionistas y publicar enlaces a la tienda virtual, es una buena forma de hacer Internet Marketing
- ✓ Utilizar las herramientas Google adSense, Google adWords y similares para publicidad en Internet.
- ✓ Tener presente que el Comercio Electrónico no es una moda pasajera, sino una tendencia cada vez más utilizada que brinda nuevas oportunidades.

A: Pérdidas en la entrega de productos

Para el envío de productos utilizar compañías de prestigio como Servientrega y/o Veloz Express a nivel nacional y DHL y/o UPS a nivel internacional, que además permiten el seguimiento de las órdenes mediante la utilización de números de guías, lo que posibilita conocer la ubicación de los pedidos en cualquier momento.

A: Falta de confianza de los consumidores en los negocios electrónicos

Se deben utilizar diferentes mecanismos para crear confianza en los clientes en lo referente a las compras por Internet. Algunas de las acciones a tomar son las siguientes:

- ✓ Informar al cliente sobre detalles de la compañía: años de experiencia, misión, visión, etc.
- ✓ Incluir testimonio de clientes satisfechos
- ✓ Responder a los e-mails de los clientes con prontitud.
- ✓ Pedir retroalimentación a los clientes y aprender de la misma.
- ✓ Resaltar en el sitio web que la tienda virtual utiliza mecanismos de seguridad que protegen a los clientes de cualquier tipo de ataque, por tanto se garantiza la culminación exitosa de cualquier transacción.

A: Falta de regulación del Comercio Electrónico en el país

Se puede investigar los avances de regulación de Comercio Electrónico en otros países y aprender de las leyes existentes para protección a consumidores y comerciantes contra el fraude.

A: Factores culturales

Se conoce que la sociedad influye en los hábitos de consumo, si se logra que un grupo de consumidores empiecen a comprar por Internet, el resto de personas se verán alentadas a utilizar este medio gracias a estos casos.

Se puede citar ejemplos de negocios virtuales en otros países para demostrar que es posible realizar transacciones seguras por Internet.

A: Falta de infraestructura: un porcentaje pequeño de la población tiene acceso a Internet

La falta de infraestructura de Internet es un problema que se tiene en Ecuador, puesto que el acceso a Internet por parte de la población es limitado, así mismo el uso de banda ancha es mínimo aproximadamente 1%.

Sin embargo últimamente se ha experimentado un crecimiento del acceso a Internet mediante el uso de tecnología ADSL de CNT y banda ancha móvil de Porta, Movistar.

Una de las ideas para mejorar la infraestructura de Internet consiste en que con el apoyo gubernamental se construyan telecentros comunitarios especialmente en sectores rurales; utilizar Internet satelital en lugares donde no es posible llegar con cableado, etc.

2.2.3.2 Aprender de la experiencia de otros negocios “punto com”

Muchos negocios “punto com” han fracasado por diversos motivos, resulta importante aprender de estas experiencias para evitar cometer los mismos errores.

- No se consideró la rentabilidad de la tienda virtual y se excedió en gastos innecesarios que a la final condujeron al fracaso del proyecto.
- La tecnología no estaba lista. Muchos proyectos de e-commerce fracasaron porque no se podía cumplir con factores claves como pagos electrónicos seguros, confidencialidad de la información, etc.
- Errores comunes en Comercio Electrónico el uso exagerado de spam, utilizar nombres de dominios gratuitos poco originales y difíciles de recordar, el uso exagerado de gráficos y animaciones que hacen pesada la página y distraen al cliente.
- Tener páginas en construcción, enlaces rotos.

2.2.3.3 Aprender de los “punto com” exitosos

Por otra parte existen casos exitosos de sitios web de Comercio Electrónico como amazon.com, se puede tomar estas experiencias positivas para implementar exitosamente la tienda virtual. Las principales recomendaciones de empresas exitosas en Internet son las siguientes:

- ✓ Tener en cuenta que el éxito en los e-mercados nunca está asegurado. Si bien el Comercio Electrónico es una idea visionaria también puede resultar riesgosa en el aspecto financiero.
- ✓ La inversión de capital debe mantenerse al mínimo cuando se trata de PyMEs (pequeñas y medianas empresas).
- ✓ Actualizar y renovar el listado de productos de la tienda virtual periódicamente. Es de vital importancia que el contenido de la tienda virtual sea renovado constantemente.
- ✓ Entrega rápida de productos. Las órdenes deben ser despachadas con prontitud, esta es una característica importante para el éxito de la tienda virtual.
- ✓ Alta disponibilidad 24/7. Garantizar a los clientes la disponibilidad del sitio en cualquier momento, en cualquier lugar.
- ✓ Categorizar lo más posible. Clasificar los productos en categorías, marcas, etc.
- ✓ Al describir productos incluir todos las características que se pueda detallar: medidas, peso, marcas, así también incluir fotografías de frente, de perfil, acercamientos, etc., con la finalidad de dar la mayor cantidad de detalles.
- ✓ En una tienda Brick and Mortar (negocio tradicional) que quiere iniciar negocios en línea no necesariamente se tiene que vender todo sino determinados productos. El tipo producto a ofrecer es un factor crítico. Hay productos que son bien vendidos en Internet y otros que resulta más difícil. Así mismo, muchas estrategias que funcionan en negocios tradicionales, no necesariamente funcionan en negocios en Internet.
- ✓ Mientras más pequeños sean los productos, más fácil será su transporte y menor el costo del mismo. Por tanto artículos pequeños son más viables que artículos grandes y pesados.

- ✓ No enfocarse solamente en ventas en el Ecuador, sino proyectarse hacia otros países, por una parte se tiene mayores clientes potenciales, y además los consumidores tiene mayor confianza y experiencia en compras por Internet.
- ✓ Publicar productos enfocados en la época del año. Por ejemplo Navidad, San Valentín, etc.
- ✓ Tener enlaces a los productos más visitados, más vendidos, destacados, etc.
- ✓ Promocionar la tienda virtual a través del negocio tradicional y viceversa.
- ✓ Utilizar la web 2.0 (redes sociales, e-bay, etc.) para promocionar la tienda virtual.
- ✓ Evitar errores tipográficos: ortografía incorrecta, evitar enlaces rotos, etc.
- ✓ Respetar derechos de propiedad intelectual, no utilizar fotografías de otras páginas.
- ✓ Publicar contenido de calidad con un diseño elegante.
- ✓ Utilizar palabras clave con etiquetas “ALT” para obtener un buen posicionamiento en los motores de búsqueda
- ✓ Automatizar la ayuda a clientes mediante la creación de la FAQ (Frecuently Asked Questions) de la tienda virtual.
- ✓ Visitar otros sitios de Comercio Electrónico exitosos y obtener ideas efectivas de promoción y ventas. Se pueden obtener valiosas estrategias, incluso de la competencia.
- ✓ Llevar estadísticas de visitas, productos más visitados, etc.
- ✓ Crear un foro de discusión.
- ✓ Entender que a diferencia de las ventas tradicionales, en las ventas online es muy relevante la identidad del comprador, por tanto la privacidad de esta información debe ser muy tomada en cuenta.
- ✓ El e-detallista más exitoso en la actualidad es amazon.com, sin embargo no tuvo ganancias en los primeros años de negocio, pero su interés estaba en el crecimiento y en aumentar el número de usuarios, las ganancias empezaron cuando el número de usuarios creció exponencialmente.

- ✓ Alentar a los amigos a visitar y ayudar a promocionar la tienda virtual, se debe tener presente que toda ayuda por más pequeña que sea es de utilidad.
- ✓ Incluir la URL de la tienda virtual en fundas, paquetes, tarjetas, etc.

Capítulo 3

Construcción de la tienda virtual

3.1 Introducción teórica

3.1.1 Aplicaciones Web

Son aplicaciones que corren bajo protocolos de Internet como el HTTP y HTTPS, y son accedidas por web browser como Internet Explorer, Firefox, Netscape, etc.

Varias tecnologías están disponibles para implementar aplicaciones web. Por una parte, CGI y PHP son comúnmente usadas, por otro lado tecnologías Java como Servlet y JSP (Java Server Pages) han ganado mucho mercado puesto que ofrecen características avanzadas. La figura 3.1 muestra la interacción de un navegador Web con una aplicación Web.

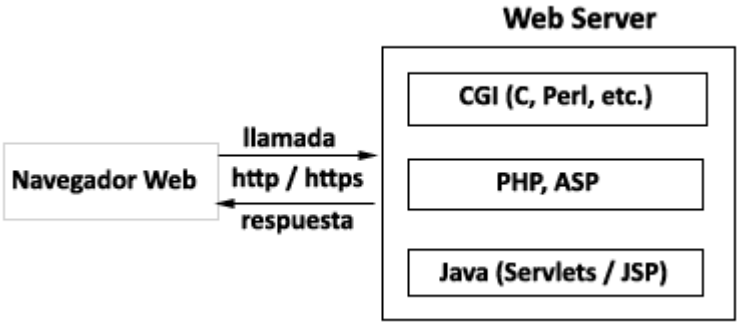

Figura 3.1 Aplicaciones Web

Las aplicaciones web tienen varias ventajas sobre las aplicaciones de propósito general, por ejemplo, puesto que las aplicaciones web requieren solamente de navegadores web para su ejecución, no se necesita instalar software adicional en los ambientes del cliente en donde se ejecuta la aplicación. Adicionalmente si un programa necesita ser actualizado, solamente se necesita actualizar el programa en el lado del servidor. Otra característica importante es que las aplicaciones web pueden ser usadas por muchos usuarios distribuidos en todo el mundo. Se puede esperar una inmensa reducción de costos por concepto de trabajos de instalación y actualización debido a que estos rubros son mucho más sencillos en comparación con las aplicaciones generales.

Debido a que las aplicaciones web pueden correr en diferentes sistemas operativos, plataformas y navegadores, una aplicación web, puede ser compartida en diferentes ambientes. Así mismo, las aplicaciones web pueden ser ejecutadas en teléfonos móviles que tienen incorporados navegadores web.

3.1.1.1 Seguridad en el desarrollo de aplicaciones web

Las transacciones que se llevan a cabo en redes abiertas como en Internet siempre están expuestas a diferentes riesgos. Existen casos en los que un atacante interfiere los datos intercambiados en la red y en los que se alteran los datos en el servidor.

Con el crecimiento continuo del uso de Internet es muy relevante tener en cuenta la seguridad cuando se diseñan sistemas que involucran aplicaciones Web.

Mediante el uso de la criptografía, autenticación y técnicas similares es factible construir sistemas seguros. Es posible prevenir la escucha de la información en la red (tapping), cifrando las transacciones de datos.

3.1.1.2 Sitios Web dinámicos

Los sitios Web dinámicos son aquellos que permiten crear aplicaciones dentro de la propia Web brindando una mayor interactividad con el navegante.

3.1.2 Servidor Web

El servidor Web es un software que permite transferir páginas Web, imágenes, formularios, archivos, de acuerdo a peticiones hechas por clientes que utilizan navegadores Web. En ambientes Unix, el servidor Web funciona como un proceso llamado deamon (demonio), el mismo que funciona en background, en cambio en ambientes Windows el servidor Web funciona como un servicio.

Los servidores Web utilizan el protocolo HTTP, no orientado a la conexión. Los servidores Web se encuentran en la capa 7 o de aplicación en el modelo OSI, el puerto predeterminado de un servidor Web es el 80. En caso de utilizar seguridad en la transmisión es el caso del protocolo HTTPS, se utiliza el puerto 443 del servidor Web.

En cuanto al desarrollo de aplicaciones orientadas a la Web, se deben considerar los lenguajes del lado del cliente (en el browser) y lenguajes del lado del servidor. En el lado del cliente tenemos los lenguajes de script que se ejecutan en el navegador: Javascript, VBScript. En el lado del servidor tenemos lenguajes que generan código HTML como PHP, JSP, ASP, etc.

3.1.2.1 HTTP Hypertext Transfer Protocol

El protocolo de transferencia de Hipertexto es utilizado en cada petición de la Web, fue desarrollado por la W3C y la IETF, se caracteriza por no guardar ninguna información de estado.

3.1.3 Servicios Web

Los Web Services o Servicios Web[5] son aplicaciones Web identificadas por una URI (Uniform Resource Identificator) cuyas interfaces pueden ser descritas y descubiertas mediante XML y soportan interacción directa con otras aplicaciones Web. Los Web Services pueden ser descritos como Web APIs (Application Programming Interface) que pueden ser accedidas sobre una red como el Internet, y ejecutadas en un sistema remoto en el cual reside el servicio solicitado. Los servicios Web disponen de operaciones que pueden ser accedidas remotamente utilizando protocolos estándar de Internet.

Entre las tecnologías utilizadas en los Servicios Web podemos destacar:

XML: describe la información a ser usada.

SOAP: empaqueta la información y la distribuye entre el cliente y el proveedor del servicio.

WSDL: Describe el servicio.

UDDI: Proporciona una lista de servicios disponibles.

La figura 3.2 describe la interacción de estas tecnologías cuando se utilizan Servicios Web.

[5] (W3 Consortium 2010)

Figura 3.2 Arquitectura orientada a servicios

3.1.3.1 XML

El XML (eXtensible Markup Language) es un lenguaje extensible de etiquetas desarrollado por la W3C, el mismo que permite definir la gramática de lenguajes específicos. Permite la compatibilidad entre sistemas para compartir la información de una manera fiable, segura y sencilla. Tiene diferentes usos en bases de datos, procesadores de texto, etc., puesto que permite transportar y almacenar información.

XML tiene similitudes con HTML, sin embargo sirven para tareas distintas, el HTML permite mostrar datos en la Web y el XML sirve para transportar datos en la Web.

El XML se caracteriza por su sencillez puesto que no utiliza estándares complejos, tan solo se trata de texto plano con etiquetas que a su vez son definidas por el usuario.

XML es independiente de la plataforma de hardware, del software y de las aplicaciones, por tanto la información es altamente disponible.

3.1.3.2 SOAP

SOAP (Simple Object Access Protocol) es un protocolo de comunicación, sirve para el paso de información entre aplicaciones. Está basado en XML para el intercambio de información de forma descentralizada en entornos distribuidos. Es independiente de la plataforma, el modelo de datos y el lenguaje de programación utilizado. La figura 3.3 describe la comunicación de datos mediante el protocolo SOAP.

Figura 3.3 Comunicación mediante SOAP

Un mensaje SOAP es un documento XML que contiene los siguientes elementos:

Envelope: identifica al documento XML como un mensaje SOAP.

Header: contiene la información de cabecera.

Body: contiene la información de llamada y respuesta.

Fault: contiene los errores y la información de estado.

La figura 3.4 describe la petición de un servicio Web por parte de una máquina cliente hacia el servidor y muestra la ubicación del protocolo SOAP en esta arquitectura.

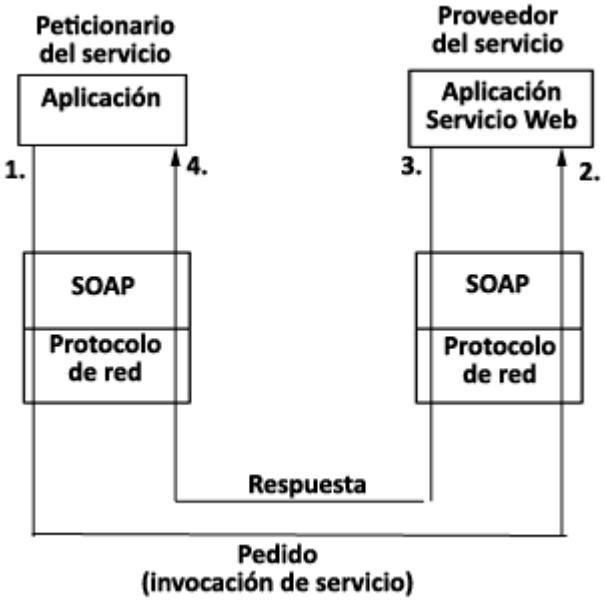

Figura 3.4 Invocación de un servicio web

3.1.3.3 WSDL

WSDL o Web Service Description Language o Lenguaje de descripción de Servicios Web, sirve para describir los servicios de red disponibles y proporciona un conjunto de puntos finales en donde se detalla la conexión. WSDL proporciona información relevante como que operaciones se pueden realizar, como se invoca el servicio y en donde se encuentra.

3.1.3.4 Consumo de Servicios Web

Existen 2 formas de utilizar los Servicios Web, la primera cuando una aplicación del lado del cliente, hace un llamado a la dirección en la que se aloja el Servicio Web y la segunda cuando un usuario realiza la petición mediante el ingreso de datos de un formulario en un servidor Web.

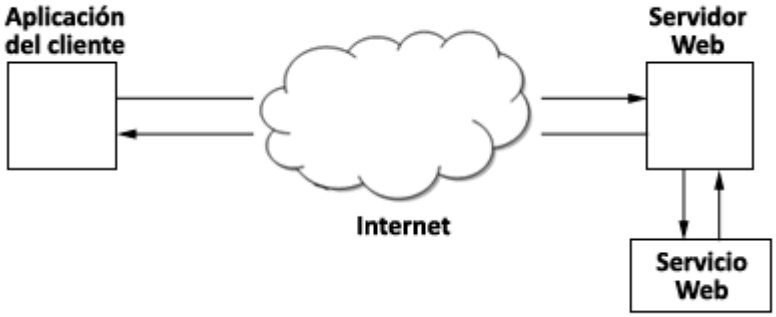

Figura 3.5 Consumo de Servicios Web

La figura 3.5 y la figura 3.6 describen la interacción entre el cliente y el (los) servidores web en el consumo de Servicios Web.

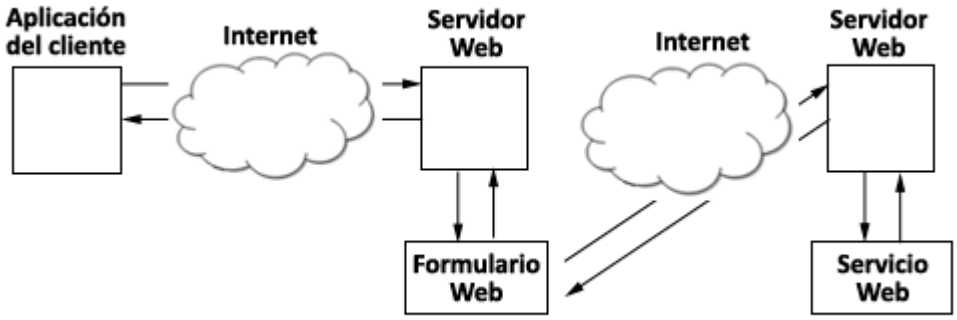

Figura 3.6 Consumo de Servicios Web (formularios)

3.1.3.5 Web Services con JAVA

JAX-WS es el acrónimo de “Java API para XML Web Services”, JAX-WS es la tecnología para construir Servicios Web y para que los clientes se comuniquen mediante XML. JAX-WS permite a los desarrolladores escribir servicios Web orientados a mensajes y orientados a RPC (Remote Procedure Call).

In JAX-WS una llamada a procedimiento remoto (RPC) es representado por un protocolo basado en XML como SOAP. La especificación de SOAP define la estructura del “envelope” (sobre) del mensaje, las reglas de codificación y las convenciones para representar las llamadas y respuestas de procedimientos remotos. Estas llamadas y respuestas son transmitidas como mensajes SOAP (archivos XML) sobre HTTP.

A pesar de que los mensajes SOAP son complejos, el API de JAX-WS esconde esta complejidad al desarrollador de la aplicación. En el lado del servidor, el desarrollador especifica los procedimientos remotos mediante la definición de métodos en una interfaz escrita en el lenguaje Java. El desarrollador también codifica una o más clases que implementan esos métodos. Los programas cliente son también fáciles de codificar. Un cliente crea un proxy (un objeto local que representa el servicio) y luego simplemente invoca métodos en el proxy. Con JAX-WS, el desarrollador no genera o convierte mensajes SOAP. El sistema JAX-WS es el que convierte las llamadas API y responde desde o hacia los mensajes SOAP.

Con JAX-WS, los clientes y servicios Web tienen una gran ventaja: la característica de Java de ser independiente de la plataforma. Además JAX-WS no es restrictivo: un cliente JAX-WS puede acceder a un servicio Web que no está corriendo en la plataforma Java.

3.1.4 Desarrollo de Aplicaciones Web

3.1.4.1 Servlets

Un Servlet[6] es un programa de Java modularizado que corre en un servidor web. El acceso a un Servlet se lo realiza a través de un navegador Web. Una petición a un navegador Web es procesada por el servidor Web e internamente enviado al Servlet, el Servlet procesa la transacción y retorna el resultado al navegador a través del servidor Web.

Las características más importantes de los Servlets son:

- Son independientes del servidor utilizado y de su sistema operativo.
- Los Servlets pueden llamar a otros Servlets, e incluso a métodos de otros Servlets. Así mismo, los Servlets permiten redireccionar peticiones de servicios a otros Servlets en una misma máquina o en una máquina remota.
- Los Servlets pueden obtener fácilmente información acerca del cliente como dirección IP, puerto que utiliza la llamada, método utilizado GET o POST, etc.
- Permiten la utilización de cookies y sesiones, de modo que se puede guardar información específica de un usuario determinado.
- Pueden realizar tareas de proxy para un applet. Debido a restricciones de seguridad, un applet no puede acceder directamente a un servidor de datos en una máquina remota, pero un servlet si puede hacerlo.
- Al igual que los programas CGI, los Servlets permiten la generación dinámica de código HTML.

El ambiente en el cual los Servlets son ejecutados, se llama Contenedor Web y está definido en las especificaciones J2EE provistas por Sun Microsystems. Las funciones de un contenedor Web no cubren las funciones de un Web server, sin embargo, muchos de los contenedores Web implementan las funciones de los servidores Web más utilizados como Apache. La figura 3.7 describe la relación entre un Servidor Web y un Contenedor Web.

[6] (Hermanos Carreño 2010)

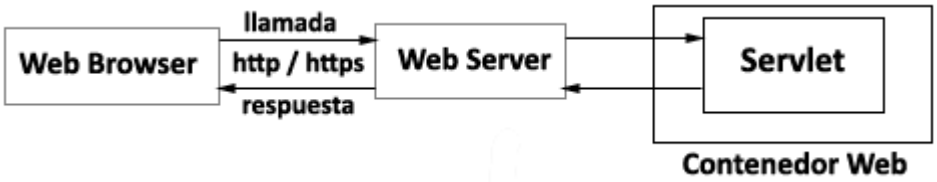

Figura 3.7 Servidores Web y Contenedores Web

3.1.4.2 CGI

La tecnología CGI (Common Gateway Interface) es una tecnología similar a los Servlets, los CGI pueden generar páginas HTML utilizando programas escritos en C o Perl, sin embargo, los CGI se ejecutan como programas externos en respuesta a cada petición de usuario, por lo que se produce sobrecarga que afecta el rendimiento del sistema cuando existen muchas peticiones que realizan llamadas a programas externos. En contraste con los CGI, toda la transacción del Servlet desde la petición hasta la respuesta es manejada como un solo proceso interno, por tanto no existe sobrecarga ocasionada con llamadas a programas como es el caso de los CGIs. Las transacciones hechas por medio de Servlets pueden ser fácilmente procesadas incluso si el número de peticiones de usuarios se incrementa, debido a que cada petición es manejada como un hilo o proceso ligero (que hacen compartición del tiempo de CPU). Adicionalmente los recursos del servidor como la memoria y el CPU son asignados eficientemente. La figura 3.8 describe la interacción entre un Servidor Web y un programa CGI.

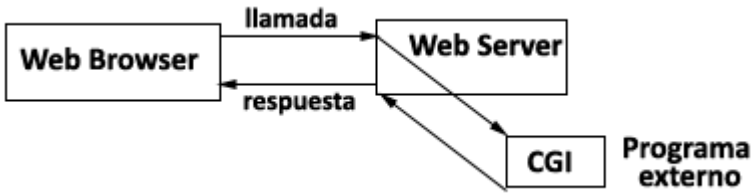

Figura 3.8 Servidores Web y CGI

Las aplicaciones Web desarrolladas con la tecnología J2EE, están compuestas por varios componentes como Servlets, archivos HTML, imágenes, archivos JSP y otros archivos de configuración. Múltiples aplicaciones Web pueden ser configuradas en un contenedor Web. La figura 3.9 describe los componentes de una aplicación Web dentro de un Contenedor Web.

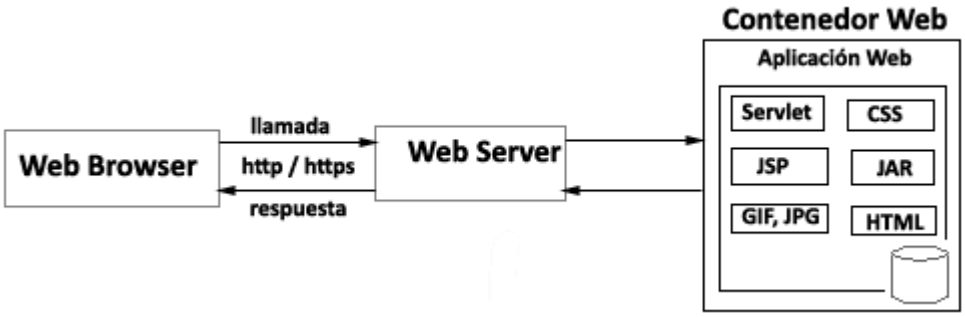

Figura 3.9 Componentes de una aplicación Web

3.1.4.3 Tomcat

Tomcat[7] es un contenedor Web Open Source distribuido por el proyecto Apache. Todos los componentes de Tomcat fueron creados usando el lenguaje Java.

La instalación de Tomcat es muy sencilla tanto en ambientes Windows como GNU Linux, simplemente se descarga de Internet y se descomprime el archivo correspondiente y se lo coloca en un directorio del sistema de archivos. Tomcat utiliza variables de ambiente para definir el directorio raíz, estas variables son CATALINA_BASE o CATALINA_HOME. En el árbol de directorios de Tomcat podemos destacar los siguientes directorios:

tomcat6/bin/ contiene programas ejecutables como startup.sh y shutdown.sh (para ambientes Linux) que inician y detienen el proceso de Tomcat respectivamente.

tomcat6/conf/ contiene los archivos de configuración de Tomcat.

tomcat6/logs/ contiene archivos en donde se guarda los mensajes de salida de Tomcat, como los mensajes de error. Uno de los archivos más importantes de este directorio es **catalina.out**

tomcat6/shared/ contiene librerías de clases utilizadas por diferentes programas.

tomcat6/temp/ es usado cuando Tomcat genera archivos temporales.

tomcat6/webapps/ es usado para guardar las aplicaciones Web cargadas en Tomcat.

[7] (The Apache Software Foundation 2010)

WEB-INF es un directorio especial de una aplicación Web que utiliza Servlets, contiene las clases de Java y archivos JAR. Los contenedores Web automáticamente configuran el directorio WEB-INF para evitar el acceso de usuarios.

Dentro del directorio WEB-INF se encuentra el archivo **web.xml** el mismo que describe la configuración de toda la aplicación Web. Este archivo es llamado "Deployment Descriptor" y su contenido es escrito en XML. (eXtensible Markup Language). Los Servlets a ser utilizados en la aplicación son descritos en web.xml.

Tomcat es activado mediante la ejecución del archivo startup.sh en ambientes Unix, que se encuentra dentro de $CATALINA_BASE/bin/ ($CATALINA_BASE es una variable de ambiente). Tomcat es configurado como un programa residente (daemon).

Para acceder a Tomcat, se debe especificar la dirección http://hostname:8080/ en un navegador Web.

En donde **hostname** es el nombre de la máquina que se utiliza como servidor y para realizar pruebas se puede utilizar el nombre localhost o la dirección 127.0.0.1 si se trabaja en la misma máquina.

El puerto 8080 es el puerto por defecto que utiliza Tomcat, sin embargo puede ser modificado debido a que otras aplicaciones pueden intentar utilizar este puerto.

Tomcat soporta funciones de servidor Web propietarios, es por esto que las aplicaciones Web pueden ser ejecutadas en Tomcat sin la necesidad de instalar otros servidores Web como Apache.

Para instalar aplicaciones Web en Tomcat simplemente se debe copiar todo el directorio dentro del directorio $CATALINA_WEB/webapps/

Para acceder a una aplicación Web ingresamos a la siguiente dirección utilizando un navegador Web:

http://hostname:8080/NombreAplicacion

Otra forma de instalar una aplicación Web es usar un archivo WAR. Los archivos WAR son archivos comprimidos que contienen el directorio completo de los archivos de una aplicación Web y son creados utilizando el comando JAR.

3.1.4.3.1 Sesiones en Tomcat

El estado de un objeto es un asunto importante en cualquier lenguaje de programación, en la programación Web este tema no varía, sin embargo se debe resaltar varias situaciones cuando se trata de manejar el estado en las aplicaciones Web.

La especificación de Servlets define la clase **javax.servlets.http.HttpSession**. Los Servlets así como JSP pueden crear una nueva sesión mediante el método: **new HttpSession()** o se puede obtener la sesión existente mediante el método **getSession()**. Si una instancia de **HttpSession** ha sido creada por una aplicación, ésta estará asociada a un identificador (ID) que mantendrá los datos del cliente, el ID creado será enviado al cliente mediante uno de los siguientes métodos: Cookies o reescritura de la URL.

Comúnmente es útil inicializar los recursos una vez cuando la sesión ha sido creada, de tal modo que puedan ser usados durante la sesión. Un buen ejemplo es la inicialización de la conexión a la base de datos y utilizarla a lo largo de la sesión. Esto es conveniente y más eficiente que crear y cerrar conexiones remotas debido a que cada petición necesita tiempo en establecer conexiones remotas.

Existen 2 métodos para inicializar recursos en una sesión. El primero tiene que ver con la verificación de que el objeto **HttpSession** sea nuevo mediante el método **isNew()**. El segundo método tiene que ver con el uso de la interfaz **javax.servlet.http.HttpSessionListener**. Cada objeto que necesite conocer si la sesión ha sido creada o destruida debe implementar la interface **HttpSessionListener** y será notificado cuando estos eventos ocurran.

Todos los objetos **HttpSession** tienen un tiempo de vida finito. Una sesión finaliza cuando ha estado inactiva por un cierto tiempo o también cuando ha sido invalidada por la aplicación. Una aplicación puede invalidar una sesión en cualquier momento invocando el método **HttpSession invalidate()**. Si una sesión no es invalidada explícitamente, automáticamente será invalidada después de un periodo de tiempo dado. El periodo de tiempo antes de la expiración de la sesión puede ser configurado en el archivo **web.xml**:

```
<session-config>
        <session-timeout>15</session-timeout>
</session-config>
```

Por defecto el tiempo de expiración de una sesión es de 15 minutos. Si se especifica un número entero en el cuerpo del elemento **<session-timeout>**, será tratado como el número de minutos que una aplicación Web tendrá que guardar la información de la sesión, si se especifica 0 o un número negativo significa indefinidamente.

Existen otros métodos para la manipulación del tiempo de expiración de la sesión:

getCreationTime(): retorna una cadena larga que representa el número de milisegundos transcurridos desde la fecha base medianoche del 1 de enero de 1970, GMT.

getLastAccessTime(): retorna la última vez que el cliente envió una petición asociada a la sesión actual.

invalidate(): invalida una sesión y quita cualquier objeto asociado a la misma.

isNew(): retorna un valor booleano que representa si la sesión ha sido recientemente creada.

setMaxInactiveInterval(int time): puede ser utilizado para establecer la cantidad de tiempo, en segundos, entre peticiones subsecuentes.

3.1.4.3.2 Cookies

Las Cookies fueron desarrolladas originalmente por Netscape Communications, para resolver el problema de mantener sesiones con el protocolo HTTP. Al utilizar Cookies, el servidor envía información al cliente para ser grabada localmente, estos bits de información son llamados “cookie”, esta cookie es utilizada para las futuras llamadas al servidor, y éste es capaz de identificar al cliente mediante los datos de la cookie. El cliente envía la cookie al servidor mediante encabezados HTTP. Las cookies necesitan que el software del cliente soporte las especificaciones de las cookies, en caso contrario no es posible su utilización. Así mismo muchos navegadores de usuarios pueden soportar cookies pero en la configuración pueden estar deshabilitadas.

Las cookies están estandarizadas por la Internet Engineering Task Force (IETF), http://www.ieft.org/rfc/rfc2109.txt.

Las cookies son pares formados por el conjunto nombre/valor, son intercambiadas mediante 2 encabezados **Set-Cookie** y **Cookie**. **Set-Cookie** va desde el servidor hacia el cliente y **Cookie** va desde el cliente al servidor. El encabezado **Set-Cookie** solo se envía una vez, cuando la sesión se establece por primera vez. Las cookies

tienen un tiempo de expiración y el cliente mantiene una copia de la cookie hasta que se cumple el tiempo de expiración.

3.1.4.3.3 Reescritura de URL

Cuando no es posible utilizar cookies, se necesita utilizar otra técnica para el seguimiento de una sesión, y lo más frecuentemente usado es un truco que utiliza el protocolo HTTP básico sin ninguna otra característica avanzada. Debido a que las aplicaciones Web necesitan entender el protocolo HTTP para realizar las peticiones al servidor Web, entonces se puede utilizar una técnica relacionada para el seguimiento de la sesión. Como su nombre lo indica reescritura de URL provee un método efectivo con el que se reemplaza la URL original, el objetivo principal es hacer que la URL contenga el identificador de la sesión.

Ejemplos de rescritura de URL son:

http://www.midominio.com?session=12345

http://www.midominio.com/session=12345

http://www.midominio.com;12345

El API de Servlets facilita la rescritura de URLs mediante el objeto **HttpServletResponse** que provee los siguientes métodos:

encodeURL(java.lang.String url): toma como parámetro una URL a codificar y retorna una cadena con el URL codificado.

encodeRedirectURL(java.lang.String url): trabaja de la misma manera que el método anterior pero codifica la URL para redireccionamiento mediante el código de respuesta HTTP 302.

3.1.4.4 Seguridad en Java

En un sistema distribuido, el código que contienen los programas, representan operaciones del negocio que son ejecutadas en uno o más servidores. Un cliente realiza una petición y ésta inicia la ejecución de código en un servidor que tiene el potencial de realizar operaciones críticas que manipulan datos sensibles del sistema. Es importante distinguir peticiones que son confiables de aquellas que no lo son. El servidor debe garantizar la seguridad basado en quien está intentando ejecutar el código, en otras palabras en la capacidad de verificar la identidad del cliente.

La seguridad de la información se complica más cuando el cliente y el servidor se comunican a través de redes públicas en donde el servidor puede ser más fácilmente manipulado. El cliente también puede necesitar la garantía de que el servidor es un elemento genuino y confiable antes de recibir información del mismo.

La seguridad en sistemas distribuidos tiene que ver en primer lugar cuando 2 partes intentan comunicarse y pueden confiar entre sí. Antes de que puedan empezar a comunicarse entre ellas, ambas partes deben saber con quién se están comunicando, para poder hacer esto cada parte debe ser capaz de probar la identidad de la otra parte, este mecanismo se lo conoce como **autenticación**. En un ambiente cliente/servidor usualmente se requiere autentificar al cliente, sin embargo existen casos en los que se necesita autentificar al servidor, es el caso del Comercio Electrónico cuando el cliente envía los datos de la tarjeta de crédito, en este caso el cliente debe ser capaz de confiar en el servidor. Existen también situaciones en las que ambas partes deben ser autentificadas. Para autentificar a una parte, ésta debe proveer un conjunto de credenciales. Las “credenciales” pueden ser de diferente tipo como pares nombre de usuario/contraseña, una huella digital, el escaneo de la retina, un certificado o cualquier cosa que identifique unívocamente a un individuo. Estas credenciales son compartidas típicamente con una autoridad de confianza para ser verificadas. Ejemplos de estas autoridades de confianza son: una base de datos con información de credenciales, un servidor Kerberos o una tercera parte como una Autoridad de Certificación como Verisign (www.verisign.com).

Una vez que una parte de la comunicación ha sido autentificada, el siguiente problema que se presenta es conocer que acciones de las solicitadas el cliente está permitido de realizar, a este proceso se lo conoce como **autorización**.

Cuando las 2 partes de la comunicación están intercambiando información, es posible que atacantes se encuentren con la posibilidad de modificar los datos mientras están en tránsito. Por tanto ambas partes deben tener la posibilidad de conocer que los datos han sido modificados. Existen técnicas que se pueden usar para realizar esta validación, la misma que en seguridad es conocida como **integridad**.

Así mismo, si suponemos que la información transmitida debe ser confidencial, es decir que un atacante aparte de no poder modificarla tampoco pueda leerla, entonces se necesita que la información sea cifrada. En lo referente al protocolo HTTP la técnica usada para manejar la encriptación es el uso del protocolo TLS (Transport Layer Security) conocido anteriormente como SSL (Secure Socket Layer).

3.1.4.4.1 Seguridad mediante declaraciones

Agregar seguridad a una aplicación puede ser una tarea complicada pero necesaria y mucho trabajo es similar sin importar el tipo de aplicación. Por ejemplo las siguientes operaciones son comunes en las aplicaciones distribuidas:

- Recibir una petición.
- Autenticar a la parte que realiza la petición.
- Verificar la autorización del cliente.
- Permitir o no el acceso.

Debido a las similitudes en las tareas de seguridad que diferentes aplicaciones necesitan hacer, es posible abstraer operaciones de seguridad de un ambiente de programación. En el caso de J2EE ciertas operaciones de seguridad no se las realiza mediante programación sino a través de declaraciones. Lo que significa que la seguridad puede ser especificada mediante configuraciones al desplegar la aplicación, por ejemplo la configuración del archivo **web.xml**, el nivel de seguridad necesario es brindado por el contenedor de la aplicación Web. Este enfoque de seguridad mediante declaraciones es flexible y relativamente fácil de usar.

Una aplicación puede escoger ignorar la seguridad mediante declaraciones y proveer sus propios mecanismos de seguridad. Típicamente una aplicación que utilice este enfoque necesita implementar filtros que se encarguen de todas las peticiones y todas las verificaciones de seguridad serán realizadas en el filtro. Un aspecto importante a tomar en cuenta acerca de la seguridad es que todas las verificaciones deben ser realizadas lo más pronto posible. Por ejemplo una aplicación no debe dejar a una petición acceder hasta puntos avanzados como la base de datos antes de decidir si el cliente está autorizado a utilizar la base de datos. Se debe tener presente que la seguridad es cara en términos de recursos utilizados, por tanto se debe verificar temprano y sólo con la frecuencia necesaria. Por ejemplo, una vez que el cliente ha sido autentificado, no se necesitará volverlo a autentificar en cada petición.

3.1.4.4.2 Seguridad basada en roles

La seguridad basada en roles es un tipo de seguridad mediante declaraciones. En la tecnología de Servlets se definen 2 términos:

Recursos: son los elementos que se necesita proteger.

Roles: son los usuarios autorizados para acceder a esos recursos.

Para entender mejor la definición de un rol, podemos pensar en un sitio Web con miles de usuarios, en el cual, habrán porciones del sitio abiertas a todos los usuarios, habrá partes en las cuales solamente serán accedidas por los usuarios que han pagado por una suscripción y habrá una porción abierta solamente a los administradores del sitio. Teóricamente sería posible verificar el acceso de los usuarios basado en las credenciales de cada usuario, sin embargo, al existir miles de usuarios esta tarea puede ser muy complicada. En su lugar, a cada usuario se le asigna un rol. Para este ejemplo habrían 3 roles: no suscriptores, suscriptores y administradores. Cada usuario será asignado a un rol el momento que explore el sitio Web, y cada recurso del sitio Web es accesible solamente a determinado rol. Se debe resaltar que rol y nombre de rol son específicos de cada aplicación y no existen roles estándar que todas las aplicaciones deban utilizar.

En el contenedor “Tomcat” el mecanismo más utilizado para manejar roles es el archivo **users.xml** ubicado en el directorio **$CATALINA_BASE/conf** ubicado en la instalación de Tomcat. Este archivo define un mapeo simple entre el nombre de usuario, el password y el rol. Es factible que un determinado usuario tenga múltiples roles. En Tomcat, las áreas en las cuales se defines los roles se denominan “realms”.

El archivo **user.xml** tiene la siguiente forma:

```
<tomcat-users>
        <user name="tomcat" password="tomcat" roles="tomcat" />
        <user name="role1" password="tomcat" roles="role1" />
        <user name="both" password="tomcat" roles="tomcat, role1" />
</tomcat-users>
```

Las restricciones de seguridad basada en roles pueden ser configuradas utilizando el elemento **security-constraint** en el archivo **web.xml**, ubicado dentro del directorio de la aplicación **WEB-INF**. Un ejemplo de esta configuración es la siguiente:

```
<security-constraint >
        <web-resource-collection>
                <web-resource-name>AplicacionSegura</web-resource-name>
                <url-pattern>/protegido/*</url-pattern>
                <http-method>GET</http-method>
                <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
                <role-name>Lector</role-name>
        </auth-constraint>
</security-constraint >
<login-config >
        <auth-method>BASIC</auth-method>
        <realm-name>Read Access</realm-name>
</login-config >
```

Las líneas de configuración anteriores son sólo una parte del contenido del archivo de configuración **web.xml** de una aplicación en el contenedor Tomcat. Con esta configuración se quiere restringir el acceso mediante el protocolo HTTP cuando se utiliza cualquiera de los métodos GET o POST y cuya URL cumpla con el patrón /protegido/*, es decir, cualquier elemento en el directorio “protegido” estará restringido.

3.1.5. JSP

JSP es el acrónimo de Java Server Pages, es una tecnología orientada a crear aplicaciones Web que se ejecuten en varios servidores Web de múltiples plataformas. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales en sintaxis Java.

El contenedor JSP (componente del servidor Web) es el encargado de tomar la página, sustituir el código Java por el resultado de su ejecución y enviarlo al cliente.

Luego de recibir una petición o llamada, un archivo JSP es traducido a un Servlet y compilado por el Contenedor Web para su ejecución. Los accesos posteriores son

manejados mediante la ejecución del Servlet. A su vez, el contenedor JSP soporta la función de actualización automática, por lo tanto, cuando un archivo JSP es modificado, el archivo es automáticamente traducido al Servlet y compilado. La figura 3.10 muestra el proceso de traducción de un archivo JSP cuando es invocado por una Navegador Web.

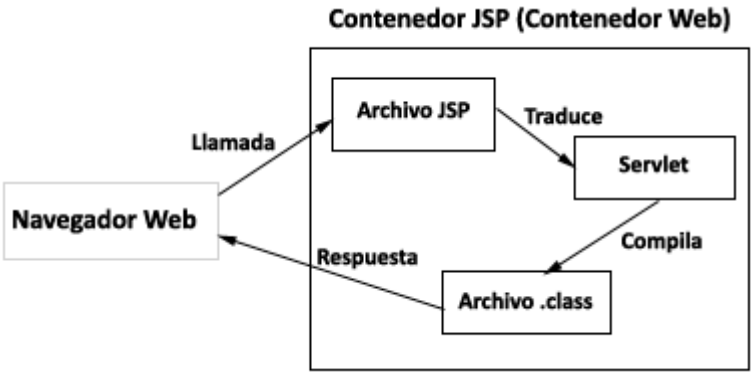

Figura 3.10 Contenedor JSP

En JSP se crean páginas con extensión .jsp que incluyen dentro de su estructura etiquetas HTML. Antes de que sean funcionales los archivos, el motor JSP lleva cabo una fase de traducción de esa página en un Servlet, implementado en un archivo .class (byte codes de Java). El Servlet generado de manera automática tiene un método **_jspService** que es el equivalente al método **service** de los Servlets "generados manualmente". En este método es donde se genera el código HTML, mediante instrucciones **println()** y donde se ejecuta el código Java insertado en la página.

JSP es el equivalente a ASP de Microsoft, ambas tecnologías permiten hacer el mismo tipo de aplicaciones. Sin embargo, JSP sigue la arquitectura JAVA de "Write Once Run Anywhere" (Escribe una vez, ejecuta donde quieras), en cambio ASP está limitada para arquitecturas basadas en tecnología Microsoft. JSP puede ser ejecutado en los servidores Web más populares: Apache, ISS, etc.

El API de JSP se beneficia de la extendida comunidad JAVA existente, por el contrario, la tecnología ASP es específica de Microsoft que desarrolla sus procesos internamente.

Tanto JSP como ASP usan una combinación de tags y scripts para crear páginas Web dinámicas, la tecnología JSP permite a los desarrolladores crear nuevos tags.

Existen 3 tipos de elementos JSP que podemos insertar en una página Web:

Código: podemos incluir código Java de distintos tipos: declaraciones de variables, expresiones, sentencias, para ser ejecutadas por el contenedor JSP.

Directivas: permiten controlar distintos parámetros del Servlet resultante de la traducción automática del JSP.

Acciones: normalmente sirven para alterar el flujo normal de ejecución de la página.

Así mismo JSP permite la utilización de objetos implícitos en su estructura. Los más importantes son: Request, Response, Out, Session, Application, Config, pageContext, Page, Exception.

3.1.5.1 JSTL

JavaServer Pages Standard Tag Library consisten en una de las herramientas más útiles de JSP, es una especificación complementaria de J2EE que definen un conjunto extendido de etiquetas JSP estándar.

La especificación JSTL 1.0 está considerada en al especificación oficial JSP (Java Specification Request) desarrollada por la JCP (Java Community Process) de Sun, con su Web http://www.jcp.org. Las especificaciones JSTL pueden ser descargadas del sitio oficial de JSTL http://java.sun.com/products/jsp/jstl.

Las funciones más importantes de JSTL incluyen: herramientas para iteración y condición, manipulación de URLs, manipulación de XML, expresiones de lenguaje, acceso a bases de datos, etc.

3.1.6 JavaBeans

JavaBeans son utilizados comúnmente con la tecnología Servlets y JSP. JavaBeans es el nombre utilizado para identificar a las clases Java que cumplen con las siguientes convenciones de programación:

- Implementan la interface java.io.Serializable.
- Tiene un constructor por defecto sin argumentos.
- Posee métodos getter y setter

JavaBeans es un objeto que guarda datos en el lado del servidor, típicamente los datos definidos en JavaBeans son de tipo privado llamados propiedades. Estas propiedades son accedidas mediante métodos getter (sirven para leer una propiedad) y setter (sirven para modificar una propiedad) que son públicos. La convención utilizada indica que para crear los métodos de una propiedad se le

agregar el prefijo “get” para obtener la propiedad y “set” para almacenar datos en la propiedad. Por ejemplo:

Propiedad: nombrecliente

Métodos: getNombrecliente()

setNombrecliente()

Los métodos getter y setter de JavaBeans no son obligatorios, sin embargo son muy importantes para la encapsulación utilizada por los JavaBeans, en la que los nombres de las propiedades del JavaBean nunca son accedidos desde fuera de la clase, lo que proporciona un nivel de abstracción importante en cuestiones de seguridad.

Los Servlets y JSP proveen una serie de funciones para utilizar y aprovechar los JavaBeans.

3.1.7 Arquitectura MVC (Modelo – Vista - Controlador)

El modelo MVC es una arquitectura de sistema que consiste en componentes discretos para el procesamiento de las peticiones, el procesamiento de los datos y el procesamiento de la presentación de los datos.

El modelo MVC fue adoptado en SmallTalk, un lenguaje de programación desarrollado por Xerox. Este modelo ha sido aplicado por muchas aplicaciones GUI (Graphic User Interface). El modelo MVC consiste en tres componentes:

Modelo: realiza el procesamiento de datos y el procesamiento de la lógica del negocio.

Vista: ejecuta el procesamiento de la presentación de los datos.

Controlador: procesa todas las peticiones del usuario, ejecuta la lógica de negocio apropiada y selecciona las páginas de presentación de datos apropiadas.

Este modelo es usado en muchas aplicaciones Web, y en especial es útil en aplicaciones que siguen el modelo de N-capas (N-tier). Por otra parte, se tiene la ventaja de que el desarrollo y el diseño pueden ser eficientemente entrelazados debido a que el diseño de pantallas y la implementación del programa son realizadas por separado.

El modelo MVC es implementado en las aplicaciones Web de Java mediante la implementación de JavaBeans para el modelo, JSP para la vista y Servlets para el

controlador. JavaBeans implementa el procesamiento de datos y la lógica del negocio, JSP despliega las páginas y el Servlet realiza principalmente el procesamiento de las peticiones. La figura 3.11 muestra la interacción entre los componentes J2EE aplicables al modelo MVC.

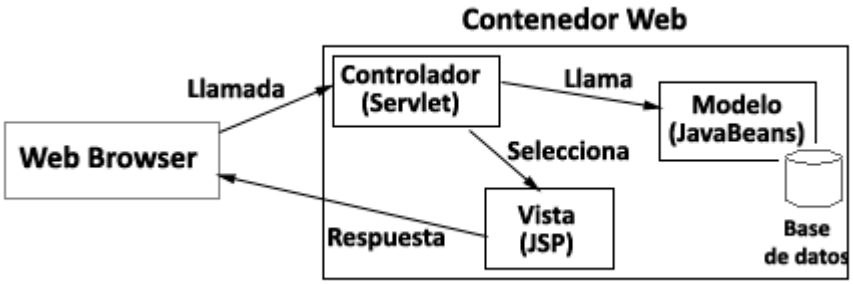

Figura 3.11 J2EE y el modelo MVC

3.1.8 JavaScript

JavaScript es un lenguaje de script (lenguaje de programación simple) desarrollado por Sun Microsystems y Netscape Communications. Fue desarrollado para agregar elementos dinámicos a las páginas web las mismas que anteriormente sólo permitían contenido estático.

Las principales funciones que se pueden llevar a cabo con JavaScript son

- Se puede controlar el flujo de un programa mediante sentencias “if” y “for”.
- Se puede construir componentes de programas mediante la orientación a objetos.
- Permite detectar eventos con el click del mouse.
- Se puede realizar el ingreso y salida de datos mediante ventanas.
- Permite verificar los datos ingresados en un formulario.
- Permite manejar páginas, marcos y ventanas.

Los programas desarrollados en JavaScript no son compilados sino interpretados por el navegador y son ejecutados en el lado del cliente. La figura 3.12 muestra el rol de JavaScript en el desarrollo de aplicaciones Web.

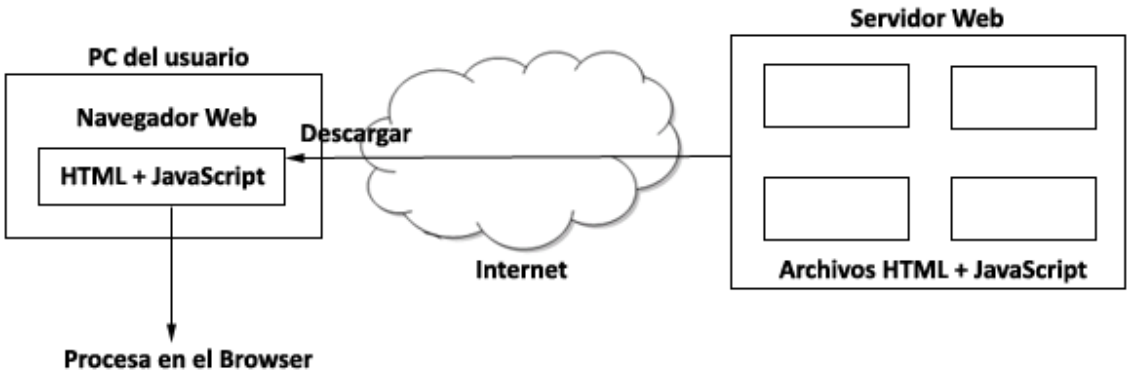

Figura 3.12 JavaScript en aplicaciones Web

JavaScript tiene similitudes con la estructura de los lenguajes C, C++, Java, sin embargo no se necesita indicar el tipo de variables.

Podemos realizar una comparación de JavaScript con el lenguaje Java resaltando las características principales, como se muestra en la tabla 3.1:

Lenguaje	JavaScript	Java
Relación con HTML	Operable con HTML	No operable con HTML
Compilación	No necesario	Necesario
Estructura del lenguaje	No existe el concepto de clases de objetos bases ni herencia de clases	Orientado a objetos
Variables	No hay limitaciones	Es estricto en verificar el tipo de variables
Desarrollador	Netscape Communications y Sun Microsystems	Sun Microsystems

Tabla 3.1 Comparación de JavaScript y Java

3.1.9 AJAX

AJAX es el acrónimo de “Asynchronous JavaScript and XML”[8], que consisten en un grupo interrelacionado de técnicas de desarrollo Web usadas en el lado del cliente con el objeto de crear aplicaciones Web interactivas. Mediante el uso de AJAX las aplicaciones Web pueden recuperar datos desde el servidor de forma asíncrona y en un proceso en paralelo sin interferir con la presentación y el comportamiento de la página Web actual. Usualmente los datos son recuperados usando el objeto **XMLHttpRequest**.

Al igual que DHTML (HTML dinámico), AJAX no es una tecnología por sí mismo, sino un grupo de tecnologías (filosofía), AJAX utiliza una combinación de HTML y CSS para el despliegue y apariencia de la información, la tecnología DOM (Document Object Model) combinada con JavaScript para desplegar e interactuar dinámicamente con la información presentada, un método para intercambiar datos asíncronamente entre el browser y el servidor, por lo que se evita la recarga de páginas, XML y XSLT para intercambio y manipulación de datos. El resumen de las tecnologías que usa AJAX lo podemos apreciar en la tabla 3.2 y su interacción la podemos ver en la figura 3.13.

Tecnología	**Función en AJAX**
XHTML y CSS	Crear la presentación de los datos basada en estándares
DOM	Interacción y manipulación dinámica de la presentación
XML, XSLT y JSON	Intercambio y manipulación de la información
XMLHttpRequest	Intercambio asíncrono de la información
JavaScript	Sirve para unir las demás tecnologías

Tabla 3.2 Tecnologías componentes de AJAX

[8] (w3schools s.f.)

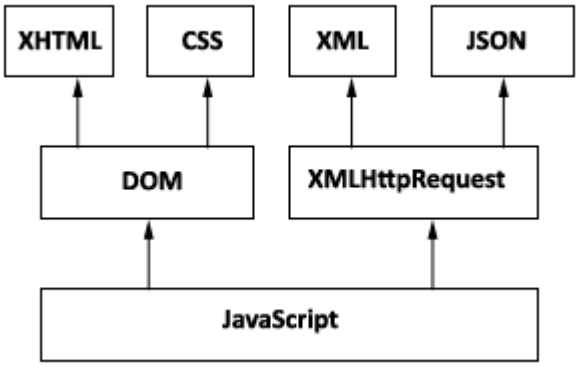

Figura 3.13 Tecnologías componentes de AJAX

Desarrollar en AJAX requiere tener conocimiento de todas y cada una de las tecnologías anteriores. En las aplicaciones Web tradicionales, las acciones del usuario en la página (hacer click en un botón o en un enlace, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una página HTML al navegador del usuario. Esta técnica tradicional para crear aplicaciones Web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en segundo plano. Las aplicaciones construidas en AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. La figura 3.14 nos permite apreciar la comparación entre las aplicaciones Web tradicionales y las aplicaciones web desarrolladas con la ayuda de AJAX.

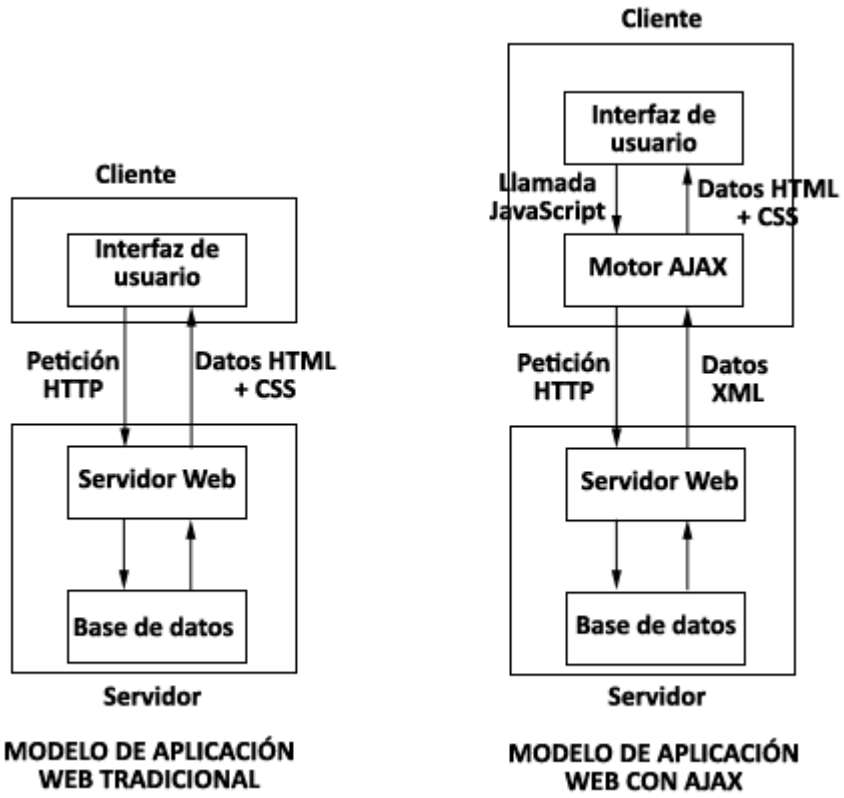

Figura 3.14 Modelos Web tradicionales vs. Modelos Web con AJAX

Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

3.1.10 DOM (DOCUMENT OBJECT MODEL)

Cuando se definió el lenguaje XML, surgió la necesidad de procesar y manipular el contenido de los archivos XML mediante los lenguajes de programación tradicionales. XML es un lenguaje sencillo de escribir pero complejo procesar y manipular de forma eficiente. Por este motivo, surgieron algunas técnicas que cumplen con este objetivo, como es el caso de DOM.

DOM o Document Object Model es un conjunto de utilidades específicamente diseñadas para manipular documentos XML. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML. Técnicamente, DOM es una

API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente.

Antes de poder utilizar sus funciones, DOM transforma internamente el archivo XML original en una estructura más fácil de manejar formada por una jerarquía de nodos. De esta forma, DOM transforma el código XML en una serie de nodos interconectados en forma de árbol. El árbol generado no sólo representa los contenidos del archivo original, sino también representa sus relaciones.

Aunque en ocasiones DOM se asocia con la programación Web y con JavaScript, la API de DOM es independiente de cualquier lenguaje de programación. De hecho, DOM está disponible en la mayoría de lenguajes de programación comúnmente empleados.

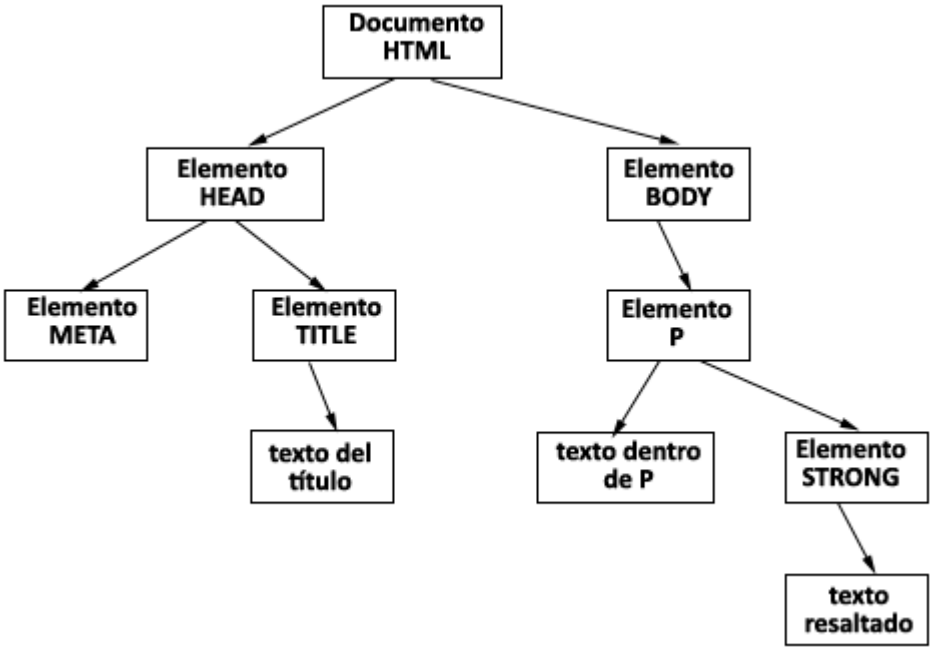

Figura 3.15 Jerarquía de nodos DOM

La ventaja de emplear DOM es que permite a los programadores disponer de un control muy preciso sobre la estructura del documento HTML o XML que están manipulando. Las funciones que proporciona DOM permiten añadir, eliminar, modificar y reemplazar cualquier nodo de cualquier documento de forma sencilla.

La primera especificación de DOM (DOM Level 1) se definió en 1998 y permitió homogeneizar la implementación del DHTML o HTML dinámico en los diferentes

navegadores, puesto que permitía modificar el contenido de las páginas Web sin necesidad de recargar la página completa.

Tipos de nodos

Los documentos XML y HTML tratados por DOM se convierten en una jerarquía de nodos (Figura 3.15). Los nodos que representan los documentos pueden ser de diferentes tipos:

Document: es el nodo raíz de todos los documentos HTML Y XML. Todos los demás nodos derivan de éste.

DocumentType: es el nodo que contiene la representación DTD empleado en la página e indicado mediante el DOCTYPE.

Element: representa el contenido definido por un par de etiquetas de apertura y cierre <etiqueta>…</etiqueta>. Es el único nodo que puede tener tanto nodos hijos como atributos.

Attr: representa el par atributo/valor.

Text: almacena el contenido del texto que se encuentra entre una etiqueta de apertura y una de cierre. También almacena el contenido de una sección de tipo CDATA.

CDataSection: es el nodo que representa una sección de tipo <![CDATA[ ]] >.

Comment: representa un comentario de XML.

Capítulo 4

Seguridad de la Información

4.1 Introducción teórica

4.1.1 Seguridad de la información

En los años recientes, las TICs (Tecnologías de la Información y Comunicaciones) han avanzado tremendamente, incrementándose la velocidad de CPU, la capacidad de memoria, la mayor cantidad de información que los dispositivos de almacenamiento pueden guardar, la velocidad de la transmisión de datos, etc. Sin bien los avances tecnológicos han facilitado nuestras vidas, nos estamos convirtiendo en dependientes de las computadoras y las redes, a tal punto que si ocurre algún incidente relacionado con la seguridad de la información, el impacto tiende a ser más grave que en el pasado.

El número de incidentes relacionados a la seguridad de la información ha crecido rápidamente en los últimos años. La razón para el incremento de estas situaciones se debe no solamente al incremento de usuarios de Internet alrededor del mundo, sino a la complejidad de los sistemas computacionales, que dan la posibilidad de crear más brechas de seguridad. Más usuarios de Internet significan más usuarios sin conocimiento de los posibles peligros de usar sistemas computacionales, y sistemas complejos significan usuarios maliciosos capaces de encontrar brechas de seguridad más rápidamente de lo que los desarrolladores de sistemas pueden manejar. Adicionalmente, ha habido un cambio radical en las técnicas utilizadas por los atacantes, dando lugar a un mayor impacto en las pérdidas, incrementando la dificultad de detectar un ataque e incrementando la dificultad de atrapar al atacante.

Existen dos categorías de seguridad, la seguridad en sentido amplio y la seguridad en sentido concreto.

La seguridad en sentido amplio tiene que ver con proteger los sistemas computacionales de todo tipo de amenazas incluyendo fuegos, terremotos, fallas eléctricas, etc.

La seguridad en sentido concreto tiene que ver con:

- Prevenir accesos no autorizados, fuga de la información, etc.

- Proteger sistemas computacionales de amenazas por acciones inapropiadas por parte de intrusos, sin considerar desastres naturales. Este tipo seguridad generalmente es llamado “seguridad de la información”.

4.1.1.1 Los 3 factores de la seguridad de la información

La seguridad de la información tiene que ver con: asegurar y mantener Confidencialidad, Integridad y Disponibilidad.[9]

Confidencialidad de la información

La información nunca debe estar accesible a usuarios sin permiso. Cualquier información almacenada o transmitida puede ser robada si un intruso accede ilegalmente a un sistema y accede a transmisiones de datos confidenciales. Datos pertenecientes a cuentas de clientes, datos de empleados, y datos referentes a estados financieros es extremadamente sensible y por tanto debería tener una mayor prioridad en la lista de bienes más valiosos de una empresa o negocio.

Integridad de las aplicaciones y la información

La información es preservada completa y precisa. La pérdida de la integridad de los datos puede llegar a ser extremadamente costosa para muchas organizaciones. La pérdida de integridad puede resultar en obtener una mala reputación, lo cual puede conducir a la pérdida de clientes y de beneficios económicos.

Disponibilidad del sistema

La información es accesible para los usuarios autorizados cuando es requerida. Cuando los recursos se vuelven indisponibles, las pérdidas de negocios pueden resultar catastróficas, especialmente en el ambiente de negocios actual, en el que más y más transacciones se basan en el uso de las redes de computadoras.

4.1.1.2 ¿Que se considera como información?

Cuando se considera la seguridad de sistemas computacionales, se tiende a pensar en proteger los datos electrónicos. Sin embargo las cosas que una organización necesita proteger incluyen no solamente los datos electrónicos, sino otros bienes que involucran información como documentos impresos, conversaciones entre empleados, y obviamente datos dentro de las computadoras y dentro de medios de almacenamiento removibles. Es por esto que la empresa debe establecer requerimientos de seguridad que incluyan:

[9] (Martínez s.f.)

- **Seguridad física** (por ejemplo: seguridad en las puertas)
- **Seguridad administrativa y operativa** (por ejemplo: capacitación sobre seguridad)
- **Seguridad técnica** (por ejemplo: contraseñas, encriptación)

Ejemplos de bienes relacionados a la información son: información del cliente, la ficha de un empleado, la información gerencial de la empresa, los datos de las transacciones del negocio, información de los productos, información de administración de los servidores, la información de la administración de la red, la información financiera, el plan de negocios, los documentos de especificaciones, los diseños (de productos o de software), etc. La tabla 4.1 muestra ejemplos de servicios de seguridad.

Ejemplos de servicios de seguridad

Servicio de seguridad	**Ejemplos en la vida cotidiana**
Autenticación	Cédula de identidad
Control de acceso	Llaves y cerrojos
Confidencialidad	Tinta invisible
Integridad	Tinta indeleble
No repudio	Firma notariada

Tabla 4.1 Ejemplos de servicios de seguridad

4.1.2 Diseño de seguridad

El objetivo principal de un diseño de seguridad tiene que ver con obtener el máximo efecto a un costo mínimo. Anticiparse a los tipos de amenazas que pudieran ocurrir de acuerdo con las características de los sistemas y tomar las medidas apropiadas. Por ejemplo:

- **Configuración del sistema:** Distribución del sitio, como está conectado con la red, la configuración del software, etc.
- **Uso del sistema:** tipos de usuarios (limitados o ilimitados), y como es usado el sistema (como los usuarios acceden al sistema.)

- **La amenaza esperada y la cantidad de daño en caso de que un evento ocurriera:** qué amenazas se puede esperar, y el resultado de comparar la cantidad de un daño potencial y el costo de las medidas de prevención.

4.1.3 Servidores Seguros

Un servidor seguro es un servidor de páginas Web configurado especialmente para establecer una conexión transparente con el cliente, de modo que la información que circule entre el cliente y el servidor viaje a través de Internet de forma cifrada para que sea legible solamente por el servidor y el cliente Web. Los servidores seguros son componentes indispensables para proteger información confidencial, por ejemplo los números de tarjetas de crédito, por tanto son imprescindibles en servicios como banca electrónica y comercio electrónico. La figura 4.1 describe un servidor seguro.

Características de los servidores seguros:

- Asegura la integridad de las sesiones con el servidor Web.
- Permiten a los visitantes de un sitio Web comunicarse de manera segura mediante un certificado digital con el que se verifica la autenticidad del sitio.
- Proporcionan cifrado de datos, autorización de servidores, integridad de mensajes y autorización de clientes para conexiones.
- Cuando se utiliza un servidor seguro, el navegador del cliente indica en la barra de direcciones el protocolo **https://,** en lugar del http://

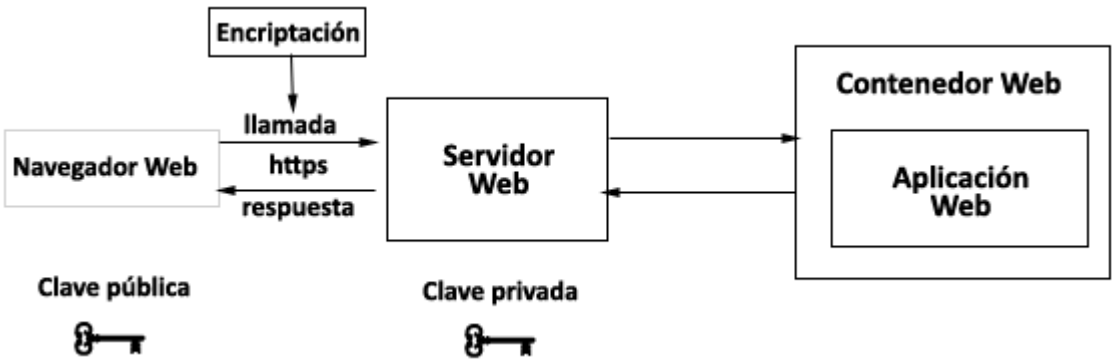

Figura 4.1 Servidores seguros

4.1.4 Autoridades de Certificación

Las Autoridades o Entidades de Certificación (CA) son empresas que se encargan de garantizar la identidad de las dos partes de una transacción. Son las organizaciones que se encargan del mantenimiento de los certificados digitales.

Las CA se encargan de autenticar a los participantes en una transacción o comunicación. El empleo de canales seguros de comunicación. Por ejemplo, cualquier persona podría suplantar a otra introduciendo correctamente los datos de su tarjeta de crédito. Es por esto, que las Autoridades de Certificación son necesarias para garantizar que cada uno es quien dice ser.

Un ejemplo de un CA es Verisign (www.verisign.com), la misma que brinda diferentes clases de certificados, dependiendo del nivel de autenticación que se necesite alcanzar.

1. **Certificados de Clase 1**, es la más baja, se emite para uso individual y puede ser conseguido en Internet, lo único que autentifica es la relación entre un nombre de usuario y una dirección de e-mail.

2. **Certificados de Clase 2**, son entregados después de comparar la información entregada por el suscriptor en determinadas bases de datos de consumidores.

3. **Certificados de Clase 3**, son los recomendados para Comercio Electrónico. En éstos certificados se verifica la autentificación mediante documentación del registro civil, incluso se exige que el usuario lleve personalmente la solicitud ante un notario,

La emisión y verificación de los certificados de autenticidad está sometida a una jerarquía de confianza, con una autoridad certificadora principal que emite certificados para los niveles anteriores.

En el esquema PKI (Public Key Infraestructure), las CA constituyen la parte confiable en la cual los participantes confían al momento de confirmar la autenticidad del otro extremo (figura 4.2).

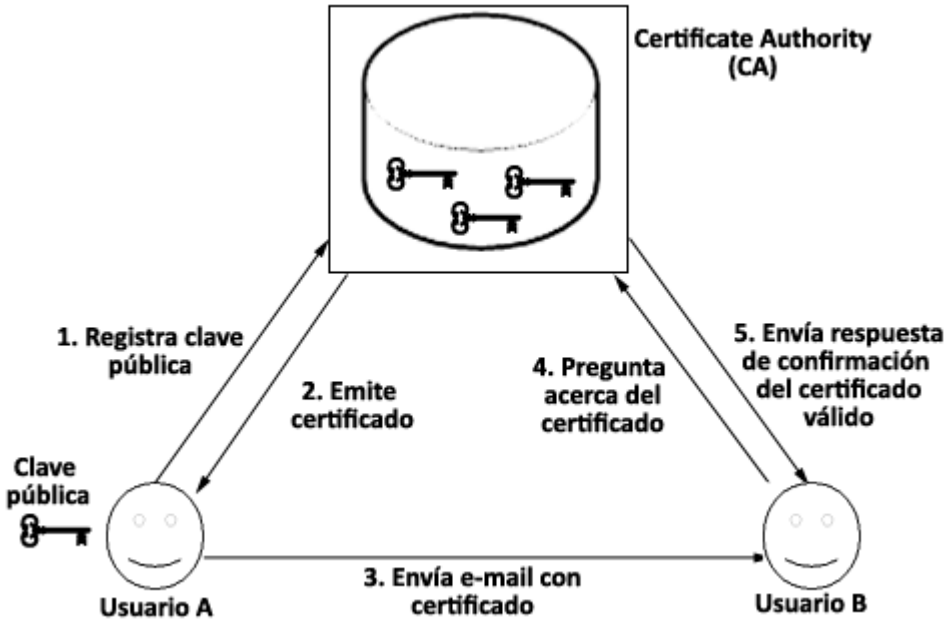

Figura 4.2 Esquema PKI

4.1.5 Criptografía

El uso de técnicas de seguridad de la información es cada vez más importante debido a aspectos como el continuo adelanto de las redes de computadoras, la masificación del uso de Internet, la transmisión de datos críticos de las empresas a través de redes inseguras. La Criptografía permite las comunicaciones seguras.

4.1.5.1 Ataques a la comunicación

4.1.5.1.1 Ataques pasivos: el atacante se limita a escuchar la comunicación, sin modificar el contenido de la misma.

4.1.5.1.2 Ataques activos: el atacante puede suplantar a una de las partes de la comunicación y modificar, eliminar, repetir, retrasar o reordenar los mensajes.

Los ataques activos se clasifican en:

- Suplantación de identidad.
- Modificación
- Degradación fraudulenta del servicio.
- Caballos de Troya.

- Encaminamiento incorrecto.

4.1.5.1.2.1 Interrupción del servicio

Consiste el ataque en que parte del sistema queda deshabilitado o no disponible. Por ejemplo: la destrucción del hardware, el corte de una línea de comunicación.

4.1.5.1.2.2 Intercepción

Una entidad no autorizada accede a parte de la información del sistema. Por ejemplo: intersección de las comunicaciones telefónicas, tapping (figura 4.3).

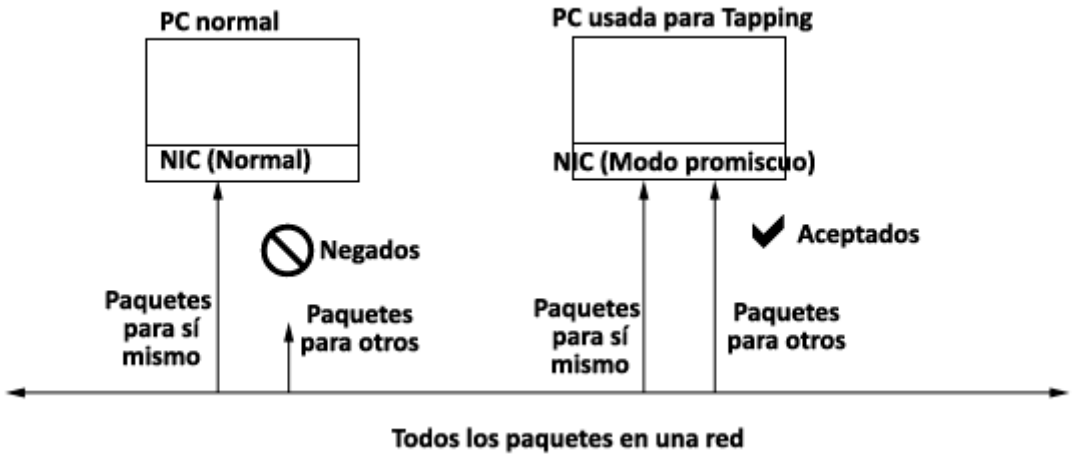

Figura 4.3 Tapping

4.1.5.1.2.3 Modificación

Una entidad no autorizada accede a la información del sistema y modifica el contenido. Por ejemplo: alteración de ficheros, alteración de programas, modificación de mensajes.

4.1.5.1.2.4 Fabricación

Una entidad no autorizada accede a la información y añade datos indebidamente y en nombre de otra entidad. Por ejemplo: una entidad envía mensajes en nombre de un usuario legítimo.

4.1.5.2 Amenazas

Podemos definir las amenazas como cualquier acción que pueda causar un daño a la red o a un sistema informático. La tabla 4.2 resume las clases de amenazas.

Clasificación de las amenazas

Clasificación		Malicia	Ejemplos
Desastres	Fuego Terremoto Tormenta Fallas eléctricas Terrorismo y guerra	No	
Problemas técnicos	Deterioro Obsolecencia Falla		
			Errores de operación
Personales	Internos	Sí	Tapping Accesos no autorizados Negación de servicio Alteración, borrado, destrucción de la información
	Externos		

Tabla 4.2 Clasificación de las amenazas

4.1.5.3 Criptología

La palabra Criptología proviene de los vocablos “cryptos” que significa oculto y “logo” que significa tratado. La Criptología se divide en 2 partes: la Criptografía y el Criptoanálisis

4.1.5.3.1 Criptografía

4.1.5.3.1.1 Historia de la tecnología de encriptación

A lo largo de la historia, el ser humano se ha visto obligado a proteger las comunicaciones trascendentales. Las comunicaciones cifradas han existido desde hace mucho tiempo atrás.

Época	Algoritmos criptográficos
Tiempos antiguos	Cifrado del César (Roma) Cifrado por sustitución
Tiempos pre-modernos	Cifrado Enigma (Alemania) Cifrado Purple (Japón)
Tiempos modernos	DES (1977 USA) cifrado de clave común RSA (1078 USA) cifrado de clave pública

Tabla 4.3 Historia de la encriptación

Si nos referimos a la Critptografía clásica, podemos destacar 3 métodos:

Cifrado por sustitución: se refiere a la conversión de códigos basados en tablas de reemplazo.

Cifrado por transposición: se refiere al cambio de posiciones de los caracteres basado en reglas establecidas.

Cifrado por interpolación: se refiere a la inserción de cadenas de caracteres suplementarias entre cadenas de caracteres.

Si se utilizan estos métodos de cifrado en combinación o repetición, la posibilidad de que la comunicación sea descifrada es menor.

4.1.5.3.1.2 Objetivos de la Criptografía.

Los objetivos de la Criptografía son:

Privacidad: un atacante que escuche la comunicación no puede obtener ninguna información acerca del contenido del comunicado.

Autenticidad: Se da al destinatario la certeza de que la comunicación proviene del origen supuesto.

Integridad: Requiere que la información sólo pueda ser modificada por las entidades autorizadas.

Verificabilidad: El destinatario sabe que la comunicación es auténtica y se le otorga la facultad de demostrarlo ante terceros.

4.1.5.4 Criptoanálisis

Es la ciencia opuesta a la Criptografía trata principalmente de analizar los criptogramas seguros. Existen diversas técnicas de análisis de criptogramas.

- Texto Cifrado conocido.
- Parejas texto claro – texto cifrado.
- Texto claro escogido (fijo y adaptativo).
- Texto cifrado escogido
- La clave o la vida

4.1.5.4.1 Ruptura de algoritmo criptográfico

Una vez realizado un ataque se puede obtener distintos tipos de ruptura del algoritmo criptográfico:

Ruptura total del algoritmo

Cuando el atacante puede descifrar cualquier criptograma con cualquier clave.

Ruptura total de la clave

Cuando el atacante ha descubierto la clave de cifrado.

Ruptura parcial

Cuando el atacante descifra un criptograma sin hallar la clave.

Información parcial

Cuando el atacante obtiene alguna información sobre el mensaje original a partir del criptograma.

4.1.5.4.2 Niveles de seguridad

En cuanto al nivel de seguridad es necesario considerar aspectos importantes en cuanto al diseño de la seguridad. Por ejemplo, el tiempo de cobertura de un mecanismo de seguridad debe ser superior al periodo de validez de la información. Por otra parte, el coste de la seguridad debe ser inferior al perjuicio de la violación.

Según los mecanismos de seguridad empleados se puede conseguir distintos niveles de seguridad.

Seguridad incondicional: Se garantiza la seguridad de la información a pesar de que el atacante tenga tiempo y recursos ilimitados.

Seguridad computacional: Se garantiza la seguridad de la información cuando los atacantes tienen tiempo y recursos limitados.

Seguridad probable: No es posible demostrar su integridad, pero el sistema aún no ha sido violado.

Seguridad condicional: El atacante carece de medios para realizar ataques que quiebren el sistema.

4.1.6 Algoritmos de seguridad

Según el tipo de clave, los algoritmos se clasifican en:

4.1.6.1 Algoritmos simétricos

Son aquellos que utilizan la misma clave para cifrar y descifrar. Ejemplos de algoritmos simétricos son: DES, IDEA, RC2, RC5, Blowfish, Safer-64. La figura 4.4 muestra la operación de los algoritmos simétricos.

Las limitaciones de los algoritmos simétricos tienen que ver con la dificultad encontrada al momento de transmitir la clave entre las partes comunicantes, este problema se acentúa para grupos con muchos usuarios

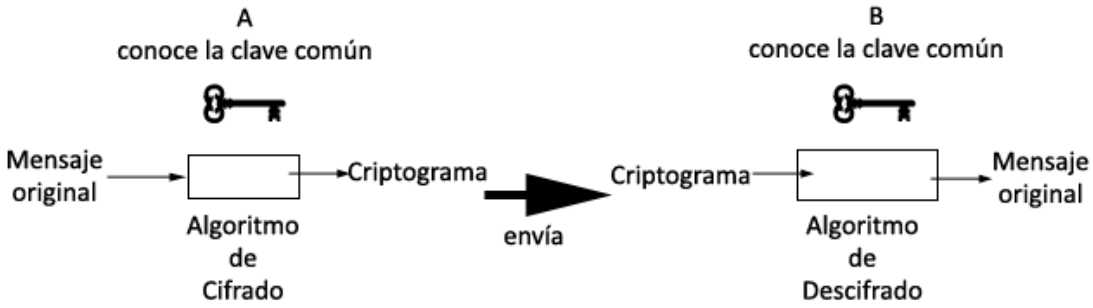

Figura 4.4 Algoritmos simétricos

4.1.6.1.1 Algoritmo Data Encryption Standard (DES)

Es un algoritmo utilizado ampliamente en comunicaciones seguras. Se basa en permutaciones, substituciones y sumas módulo 2. Emplea una clave de 56 bits y opera con bloques de datos de 64 bits.

Los ataques al DES consisten:

- Fuerza bruta
- Ataque con texto claro conocido.
- Ataque con texto claro elegido.
- Criptoanálisis diferencial. Se realizan comparaciones XOR de 2 textos en claro elegidos y sus correspondientes criptogramas.
- Criptoanálisis lineal. Tiene que ver con la obtención de un método lineal que represente la relación entre algunos bits del mensaje en claro, con el criptograma y la clave.

4.1.6.1.2 Algoritmo International Data Encryption Algorithm (IDEA)

Es considerado uno de los mejores algoritmos simétricos de la actualidad. Trabaja con bloques de 64 bits de longitud, al igual que el DES, pero utiliza una clave de 128 bits. Se utiliza el mismo algoritmo tanto para cifrar como para descifrar. Se basa en los conceptos de confusión y difusión, utilizando compuertas XOR.

4.1.6.1.3 Algoritmo Advance Encryption Standard (AES)

Es el posible sucesor del DES. El tamaño de la clave debe ser de al menos 128 bits, al igual que el tamaño del bloque de cifrado. Se caracteriza por ofrecer una buena combinación de seguridad, eficiencia, sencillez y flexibilidad.

4.1.6.2 Algoritmos asimétricos

Utilizan dos claves distintas, la una pública y la otra secreta. La clave privada sólo la posee el receptor y la utiliza para desencriptar. La clave pública la posee el receptor, pero se la pasa al emisor para que la utilice al momento de cifrar su mensaje. Las figuras 4.5 y 4.6 describen el funcionamiento de los algoritmos asimétricos.

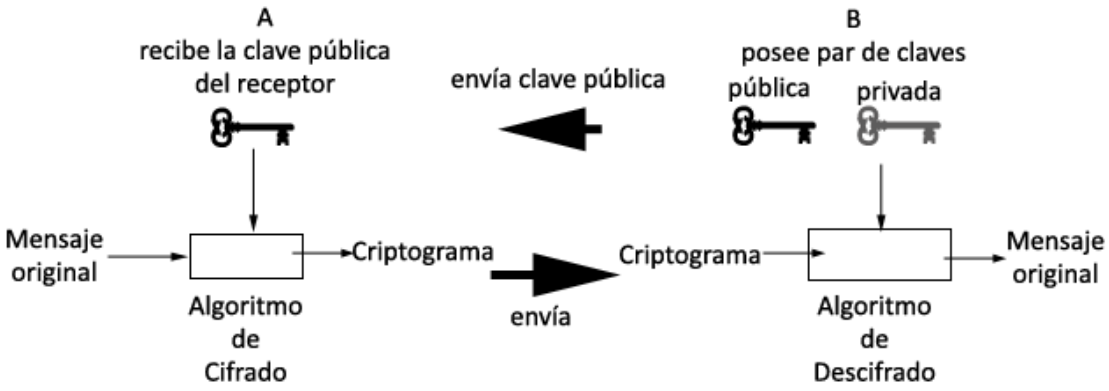

Figura 4.5 Algoritmos asimétricos

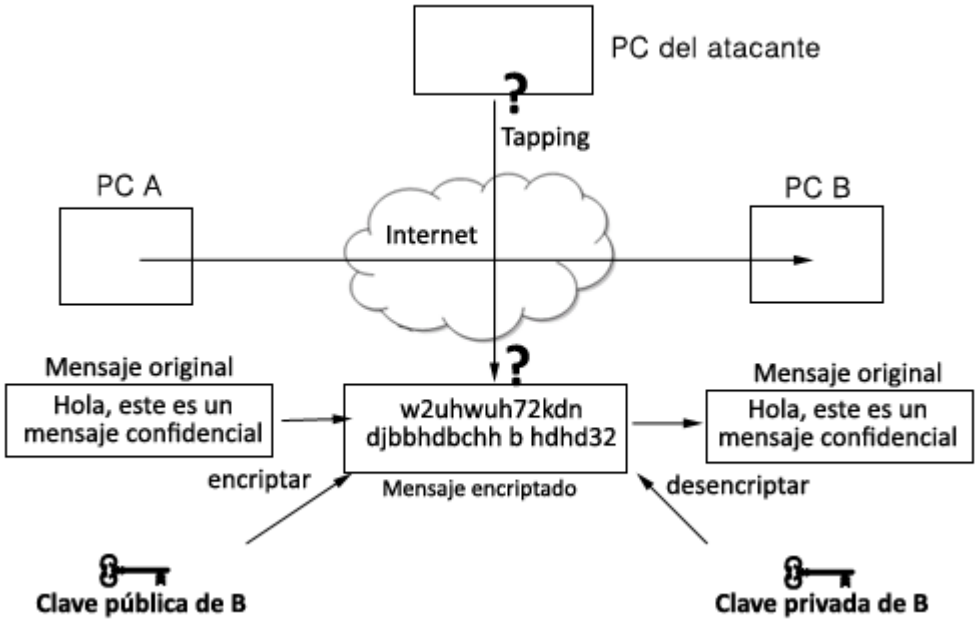

Figura 4.6 Algoritmos asimétricos en la Web

Los algoritmos asimétricos son más seguros puesto que, a pesar de que un intruso consiga la clave pública, no será capaz de encontrar la clave privada a través de la clave pública para poder descifrar el mensaje. El principal inconveniente de los algoritmos asimétricos es que su implementación resulta muy costosa computacionalmente. Ejemplos de algoritmos asimétricos: RSA, Elgamal.

En cuanto a la descripción de los algoritmos asimétricos podemos acotar que se definen 2 algoritmos matemáticos E y D los mismos que son públicos y dependen de ciertas claves. Dado un mensaje M, se tiene que

D (E (M)) = M

Es decir al aplicar el algoritmo E sobre el mensaje obtenemos el texto cifrado C y al aplicar el algoritmo D sobre el texto cifrado C, D (C), obtenemos el mensaje original.

Se tiene que es difícil hallar el algoritmo D a partir de E. Estos algoritmos se basan en ciertas funciones unidireccionales, función trampa.

Cada comunicante tiene 2 claves una pública y otra privada. La clave privada no es posible de ser calculada a partir de la clave pública.

Los algoritmos asimétricos deben cumplir las condiciones indicadas en la tabla 4.4:

A partir de estos datos	**Obtener**	**Dificultad**
Clave pública, mensaje original	Texto cifrado	Fácil
Clave privada, texto cifrado	Mensaje original	Fácil
Clave pública	Clave privada	Difícil
Clave privada, texto cifrado	Mensaje original	Difícil

Tabla 4.4 Condiciones en algoritmos asimétricos

Para representar la comunicación entre 2 usuarios A y B, utilizando un algoritmo de seguridad asimétrico, utilizaremos el siguiente esquema

Usuario A tiene **clave pública A** y **clave privada A**

Usuario B tiene **clave pública B** y **clave privada B**

Algoritmos matemáticos E y D

Operación	**Cifrado**	**Descifrado**
Confidencialidad	Texto cifrado = E aplicando clave pública B a mensaje original	Mensaje original = D aplicando clave privada B a texto cifrado
Autenticidad y firma digital	Texto cifrado = E aplicando clave privada A a mensaje original	Mensaje original = D aplicando clave pública A a texto cifrado

Tabla 4.5 Operación de los algoritmos asimétricos

La Criptografía Asimétrica permite obtener los objetivos de seguridad como confidencialidad y autenticidad en una comunicación. Permite el uso de la firma digital sin la intervención de una entidad certificadora.

En cuanto a la transmisión de las claves públicas, se tiene la ventaja de que el crecimiento del número de claves es lineal con relación al número de usuarios.

Función trampa

La función trampa utilizada en los algoritmos asimétricos hacen referencia a funciones que son fáciles de calcular en un sentido, pero la función inversa es muy difícil de calcular sino se tiene cierta información adicional. La información adicional necesaria también es difícil de obtener.

A continuación se detallan los métodos utilizados con sus respectivos artificios matemáticos.

Método	**Basado en …**
Diffie-Hellman	Logaritmo discreto
Massey-Omura	Logaritmo discreto
Elgamal	Logaritmo discreto
RSA	Factorización
Miller	Logaritmo elíptico
Probabilístico	Residuosidad cuadrática

Tabla 4.6 Artificios matemáticos utilizados en algoritmos asimétricos

4.1.6.2.1 Algoritmos RSA

Es el algoritmo asimétrico más fácil de implementar y es considerado uno de los algoritmos asimétricos más seguros. Su nombre proviene de sus creadores Rivest, Shamir y Adleman. Se basa en la dificultad para factorizar números grandes, puesto que las claves se calculan a partir de un número que se obtiene como resultado del producto de 2 números primos grandes. Es el algoritmo utilizado en el SSH (Secure Shell Client).

Comparación de los algoritmos criptográficos simétricos y asimétricos

Tipo de algoritmo	Simétrico	Asimétrico
Claves	Clave de cifrado es igual a clave de descifrado	Clave de cifrado es distinta de clave de descifrado
Manejo de claves	Diferentes claves comunes son necesarias para diferentes usuarios	Manejo simple de claves debido a que una clave privada es usada para múltiples comunicaciones
Entrega de claves	Secreta	Clave pública puede ser publicada
Velocidad de procesamiento	Rápida	Lenta
Ejemplo	DES, AES	RSA

Tabla 4.7 Comparación de los algoritmos simétricos y asimétricos

4.1.7 Clasificación de los algoritmos de encriptación según el bloque de datos

Según el bloque de datos, los algoritmos simétricos se dividen en algoritmos de flujo y bloque[10]

4.1.7.1 Cifrado en flujo

El tamaño del criptograma coincide con el tamaño del texto original. El cifrado en flujo es considerado un dispositivo con memoria debido a que en la encriptación se necesita recordar el estado del elemento anterior. Ejemplos de algoritmos de flujo son: Jennings, A5.

4.1.7.2 Cifrado en bloque

El texto es dividido en bloques de tamaño fijo, si faltan caracteres, es necesario añadir un relleno. El cifrado en bloque es considerado un dispositivo sin memoria porque en la encriptación no es necesario recordar el estado del elemento anterior. Ejemplos de algoritmos de bloque son DES, IDEA.

[10] (Forné s.f.)

Los símbolos de un mensaje son agrupados en bloques de 2 o más elementos. Cada símbolo se cifra de forma dependiente de los otros símbolos que pertenecen al mismo bloque.

4.1.7.2.1 Elementos del cifrado en bloque simétrico

Transformación inicial

Es una función que permite aleatorizar los datos de la entrada para ocultar bloques, con lo que dificulta el criptoanálisis.

Iteraciones intermedias

Son funciones no lineales complicadas. Pueden ser unidireccionales como el algoritmo DES, o no como en los algoritmos IDEA, RC5.

Las funciones no lineales consisten en una sola operación compleja conformada por una sucesión de varias transformaciones simples. Los datos se enlazan mediante la operación lógica XOR con datos procedentes de la transformación inicial o de iteraciones precedentes.

La operación de transformación final permite que las operaciones de cifrado y descifrado sean simétricas.

4.1.8 Herramientas criptográficas

4.1.8.1 Función de Hash

Una función de Hash H, transforma un mensaje M de longitud variable y lo convierte en una cadena de longitud fija. La función de Hash H(M), es relativamente fácil de calcular para cualquier mensaje dado. La función de Hash tiene la característica de que es computacionalmente imposible encontrar dos mensajes cuya función de Hash sea igual. Las funciones de Hash más utilizadas son MD2, MD4, MD5, SHA.

Las aplicaciones de la función de Hash tienen que ver con la firma digital de documentos y la verificación de claves públicas.

4.1.8.2 Firma digital

La firma digital es una técnica que usa un procedimiento inverso al cifrado de clave pública, y su objetivo es el de detectar la alteración de un mensaje o documento. Con la firma digital, el emisor transmite los datos cifrados con su clave privada y el

receptor la descifra con la clave pública del emisor. Con la firma digital no se puede prevenir que los datos sean vistos por una tercera parte, puesto que cualquiera puede descifrar el mensaje con la clave pública del emisor, sin embargo, como esta tercera parte, no posee la clave privada del emisor, no puede tomar la identidad del emisor para enviar datos al receptor. Por tanto, si el receptor no puede descifrar los datos usando la clave pública, el receptor sabrá que el mensaje fue enviado por una tercera parte o fue alterado.

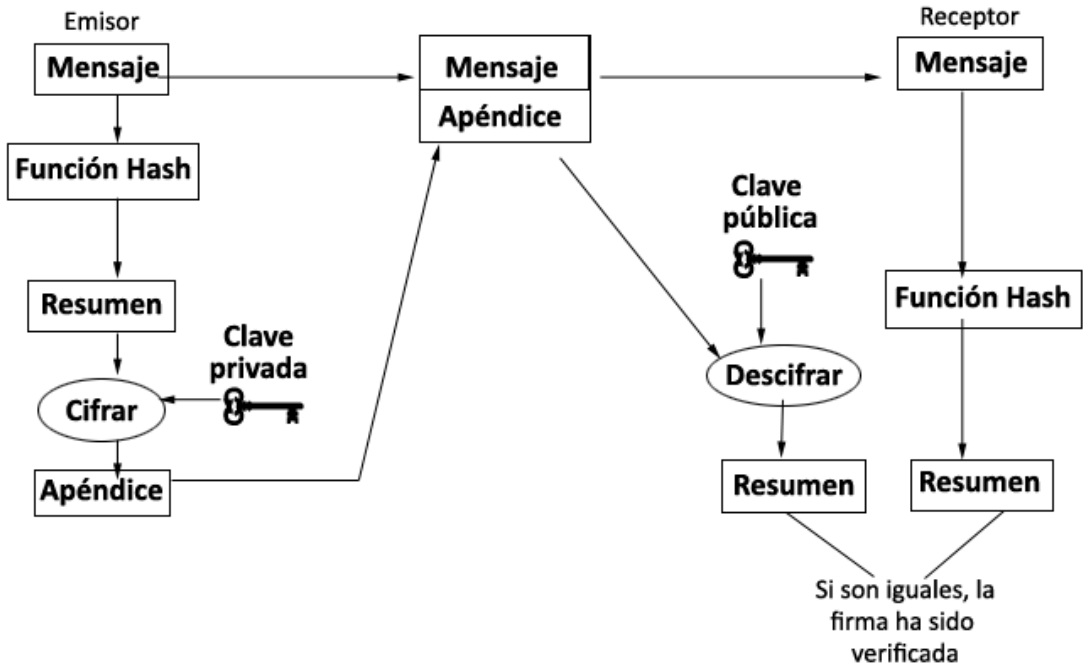

Figura 4.7 Firma digital

Características de la firma digital

- La firma digital debe ser auténtica.
- La firma digital no puede ser falsificada.
- La firma digital no puede ser reutilizada.
- El documento firmado no puede ser alterado.
- La firma no puede ser repudiada por el firmante.

Verificación de la firma digital

Protocolo de verificación de la firma digital implementado con funciones de Hash.

1. A calcula el Hash de un documento.
2. A cifra el Hash obtenido con su clave privada.
3. A envía el documento y el hash cifrado a B.
4. B calcula el Hash del documento. B descifra el hash firmado, utilizando la clave pública de A. Si ambos coinciden, la firma es válida.

Funciones de la firma digital

- Integridad
- Autenticación
- No repudio

4.1.8.3 Certificados digitales

Es un fichero digital intransferible y no modificable emitido por una autoridad de certificación CA, que da fe de la vinculación de una clave pública con un individuo o entidad. Uno de los formatos más utilizados es el presentado por la Unión Internacional de Telecomunicaciones. Los certificados digitales están compuestos por los siguientes elementos:

- Versión
- Número de serie.
- Identificación del usuario – nombre de la persona o entidad
- Clave pública
- Nombre CA (Autoridad de Certificación).
- Periodo de validez – fecha de expiración
- Firma digital de la entidad emisora

4.1.8.3.1 Funciones de los centros de verificación

Las funciones de los centros de verificación tienen que ver con:

- Emisión de certificados para nuevos usuarios.
- Rutinas para modificar o dar de baja un certificado.
- Generar listas de revocación.
- Comunicarse con otros centros de certificados.

4.1.8.3.2 Componentes de Infraestructura de clave pública

Al trabajar con infraestructuras de clave pública son necesarios los siguientes componentes:

- Certificados digitales X.509v3.
- Autoridad de Certificación (CA).
- Autoridad de Registro (RA).
- Autoridad de Validación (VA).

Los certificados digitales pueden usarse con los protocolos SSL/TLS, sobres digitales S/MIME, PGP.

4.1.9 SSL / TLS

Secure Socket Layer es un protocolo diseñado para proveer Privacidad e Integridad de la información en una comunicación.

El protocolo SSL fue el antecesor del TLS, este último se convirtió en el estándar RFC2246.

En el modelo TCP / IP el protocolo SSL / TLS está ubicado por debajo de la capa de aplicación y por encima de la capa de transporte (figura 4.8), es por esto que se pueden ejecutar diversas aplicaciones que utilicen SSL / TLS como la navegación Web y el correo electrónico.

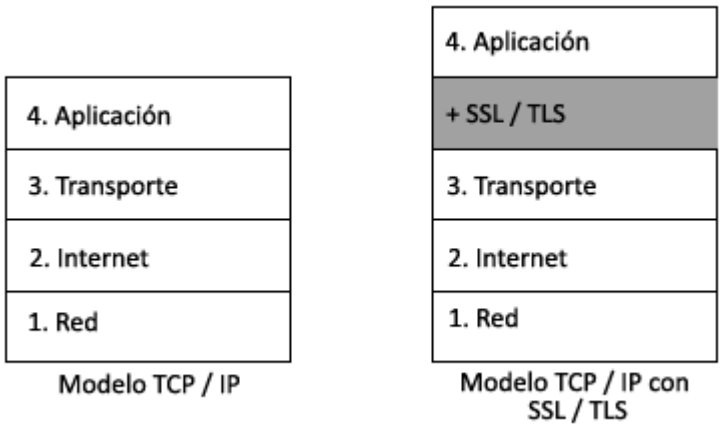

Figura 4.8 SSL/TLS bajo el modelo TCP/IP

El protocolo TLS se ha diseñado para evitar que las aplicaciones que funcionan bajo redes sean espiadas, existan intromisiones o falsificaciones, así mismo es independiente del protocolo de la capa de aplicación, por tanto protocolos de alto nivel pueden colocarse transparentemente sobre el protocolo TLS (Figura 4.9).

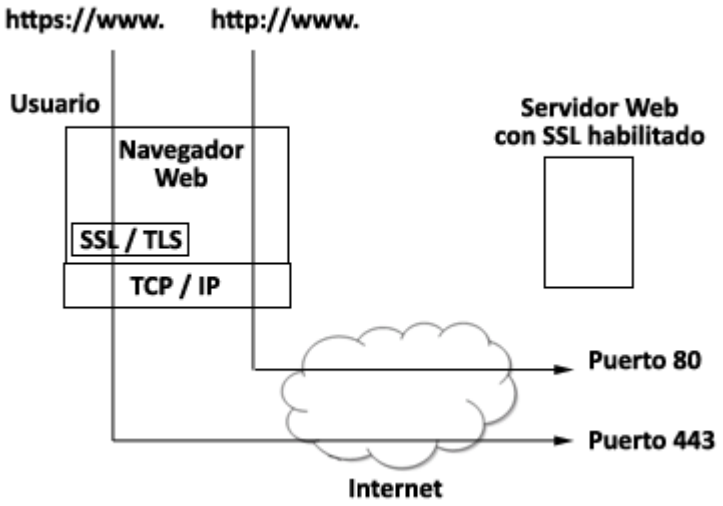

Figura 4.9 TLS y aplicaciones

4.1.9.1 Fases del protocolo TLS

Las fases del protocolo TLS son:

1. **Handshake**

   Durante el handshake se negocia el algoritmo de cifrado, las claves y se realiza la autenticación del servidor y del cliente.

2. **Transmisión de datos de la aplicación.**

   Una vez terminado el handshake, se inicia la transmisión de datos de la aplicación. Para ello se cifran todos los datos utilizando las claves de sesión negociadas durante el handshake.

Existen restricciones de exportación para las versiones del protocolo SSL/TLS. Dentro de los Estados Unidos se comercializa una versión de 128 bits, fuera de Estados Unidos se exporta una versión que utiliza claves de 40 bits.

4.1.9.2 Componentes del protocolo TLS

El protocolo TLS está compuesto por dos capas:

- TLS record protocol
- TLS handshake protocol.

4.1.9.2.1 TLS record protocol

Este protocolo se encuentra en el nivel más bajo, es soportado sobre TCP. Se puede utilizar TLS record protocol para encapsular varios protocolos de nivel más alto.

Privacidad en TLS record protocol

Se usa criptografía simétrica para cifrado de datos, por ejemplo los algoritmos DES, RC4, etc.

Las claves para el cifrado simétrico se generan de forma única para cada conexión.

Se basa en una clave secreta negociada durante el TLS handshake protocol.

Integridad en TLS record protocol

El transporte del mensaje incluye verificación de integridad del mensaje. El cálculo del código de autenticación del mensaje MAC (Message Authentication Code) se lo realiza mediante funciones de Hash utilizando algoritmos como SHA, MD5, etc.

4.1.9.2.2 TLS handshake protocol

Permite autenticar el servidor y el cliente, para ello negocia el algoritmo de cifrado a utilizar y las claves criptográficas antes que el protocolo de aplicación empiece a transmitir o recibir datos.

La negociación de la clave secreta compartida es segura, la clave secreta negociada no está disponible para los atacantes. La negociación de la comunicación es confiable, el atacante no podrá modificar la negociación sin ser detectado por las partes comunicantes.

Finalmente, la figura 4.10 resume el funcionamiento completo del protocolo TLS.

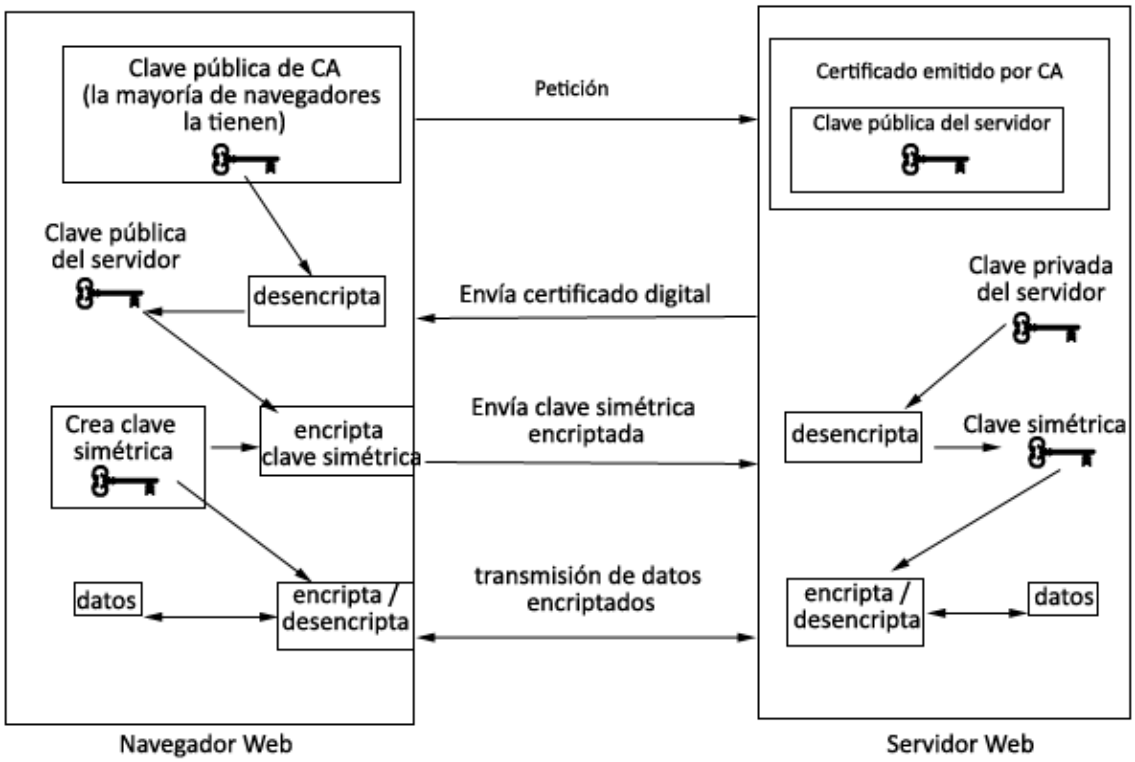

Figura 4.10 Operación de TLS

4.1.10 Encriptación de correo electrónico

Un mensaje de correo electrónico atraviesa varios servidores de e-mail antes de alcanzar el servidor de e-mail de destino. En su camino hacia su destino final, existen muchas posibilidades de ser interceptados por terceros (usuarios

maliciosos). Especialmente, los administradores de red pueden revisar fácilmente lo que está escrito en los mensajes de e-mail de sus usuarios. Para resguardar información confidencial se utiliza técnicas de encriptación de e-mails.

4.1.10.1 S/MIME

S/MIME es el acrónimo de Secure/Multipurpose Internet Mail Extensions. Cuando los datos son encriptados con el formato MIME, varios tipos de información como audio e imágenes pueden ser encriptados. S/MIME es soportado por programas como Microsoft Outlook Express.

4.1.10.2 PGP

PGP es el acrónimo de Pretty Good Privacy. Existe una versión de PGP no comercial, ampliamente utilizada, y que puede ser descargada de Internet. PGP usa sus propias claves y utiliza ambos métodos de cifrado de clave común secreta y clave pública.

PGP cifra el mensaje utilizando un algoritmo simétrico, puesto que son más rápidos que los asimétricos. Para ello utiliza una clave generada aleatoriamente y posteriormente codifica la clave mediante un algoritmo asimétrico haciendo uso de la clave pública del destinatario. Cada clave aleatoria sirve únicamente para una sesión, con esto se consigue que si un intruso logra descifrar una clave, no la podrá utilizar en mensajes transferidos en las siguientes sesiones.

La diferencia entre S/MIME y PGP es como garantizar las claves públicas. S/MIME utiliza CA (Autoridades de Certificación). PGP no utiliza CA pero utiliza la firma de otra persona la misma que es confiable y es utilizada en lugar de CA.

4.1.11 Encriptación de archivos

Incluso si la información es cifrada en una red, después de ser descifrada es guardada en el servidor, CDROMS, etc., por tanto, podemos afirmar que es expuesta a amenazas internas.

Por ejemplo, los computadores personales en una oficina contienen información confidencial. Los medios de almacenamiento removible como memorias flash, CDs, DVDs, diskettes constituyen una fuente de fuga de información debido a que pueden ser robadas o resultar extraviadas, exponiendo la información que almacenan. Así mismo en las organizaciones sólo usuarios autorizados tienen acceso a información confidencial como información financiera, nóminas de empleados, roles de pago,

estados contables, etc., y otros usuarios tienen acceso restringido a ese tipo de información. Con el objeto de proteger la información confidencial de una organización se puede utilizar técnicas de encriptación de archivos.

4.1.12 Control de Acceso

Los niveles de confidencialidad, integridad y disponibilidad difieren en las actividades de una organización. Para garantizar confidencialidad, integridad y disponibilidad es necesaria una clasificación e implementación del control de acceso. Si los requerimientos de un negocio por control de acceso no están claros o es difícil su implementación, entonces pueden ocurrir incidentes de seguridad. Por ejemplo, si el acceso a la base de datos de clientes, no está limitada a usuarios autorizados, los riesgos de robo de información o de alteración de la misma se incrementan.

4.1.12.1 Control de acceso para servidores Web

El control de acceso para servidores Web ha sido equipado con controles básicos. Cuando se realiza un intento de acceso a cierta vista Web con restricciones, se implementa un control de acceso básico que constituye el uso de un nombre de usuario y una contraseña.

4.1.12.1.1 SSO (Single Sign On)[11]

SSO es un ejemplo de Control Acceso Integrado, el mismo que permite realizar la autenticación en un servidor una sola vez y luego el usuario acceder a múltiples servidores de la misma red. El uso de un nombre de usuario y contraseña se vuelve complicado debido a la expansión del sistema, además evita que el nivel de seguridad se degrade.

[11] (NTT DoCoMo 2005)

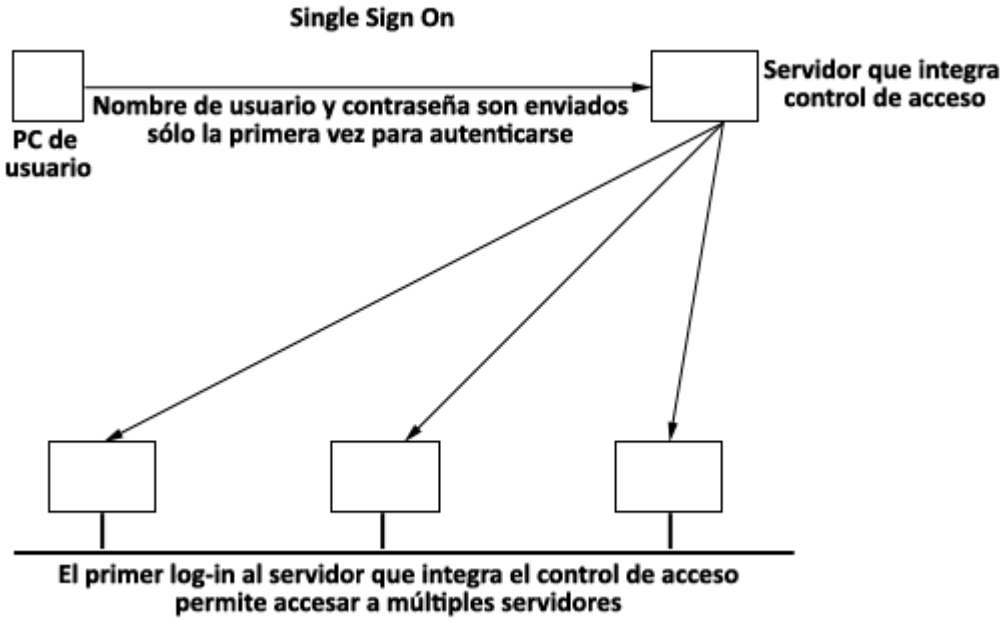

Figura 4.11 Single Sign On

4.1.13 Autenticación

Una gran cantidad de información es transmitida entre organizaciones como envío y recepción de e-mail, acceso a recursos internos, intercambio de datos, etc., todos estos llevados a cabo en ambientes Internet / intranet. Incluso si se utiliza encriptación, este ambiente todavía estará expuesto a amenazas como acceso por parte de personas no autorizadas y envío / recepción de datos como resultado de robo de identidad.

Por tanto, es recomendable la utilización de métodos de autenticación, los mismos que incluyen el uso de objetos externos como tarjetas de identificación y llaves, características biológicas como la huella digital, etc. Mientras se tenga una conexión a Internet o la intranet, el objeto físico como medio de autenticación no está siempre disponible a todos los usuarios, por tanto la contraseña se ha convertido en el medio de autenticación más usado.

Con el objeto de equipar mejor el ambiente de autenticación, el ingreso de la contraseña “una sola vez”, puede ser incorporado debido a que mejora la seguridad. Se puede también utilizar un servidor de autenticación para facilitar el manejo de contraseñas.

4.1.13.1 Método Reto / Respuesta (Challenge / Response method)

Cuando se utiliza el método reto / respuesta, el usuario prepara una computadora pequeña llamada HHA (Hand-Held Authenticator). Para que el usuario acceda al sistema, primero accede al servidor y recibe un código de reto del mismo. Cuando el usuario ingresa el código recibido al HHA, este dispositivo crea una contraseña (respuesta) de acuerdo a la lógica programada. Finalmente esta contraseña es utilizada para el ingreso al sistema.

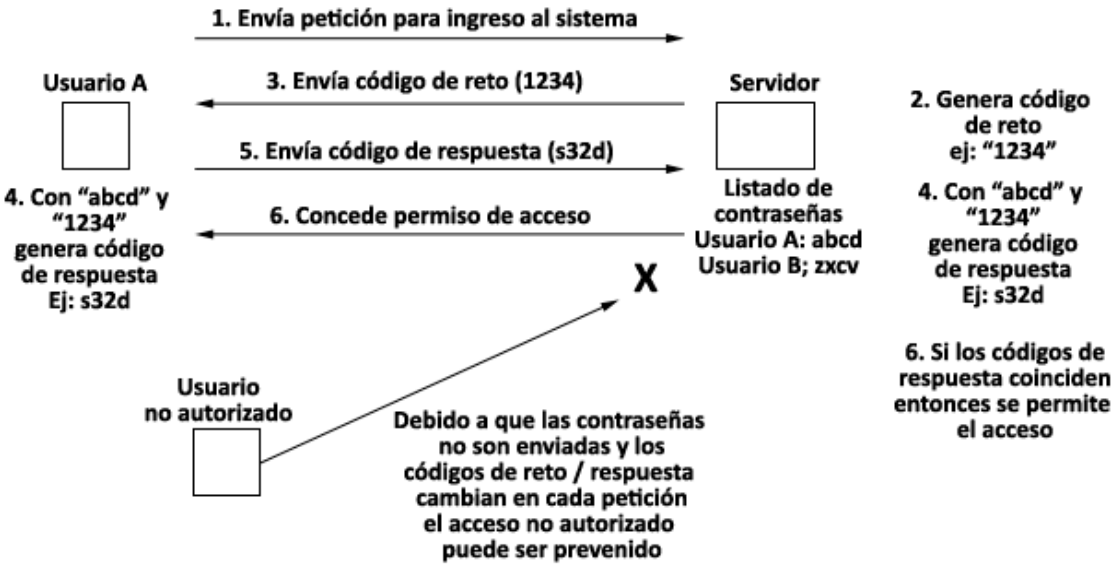

Figura 4.12 Challenge / Response Method

4.1.13.2 Método de autenticación síncrono de tiempo (Time Synchronous Authentication Method)

El usuario del cliente utiliza la contraseña mostrada en un testigo para realizar el log-in. Las contraseñas están sincronizadas en el tiempo para cada usuario entre el servidor y el testigo. Si las contraseñas concuerdan, el servidor permite el acceso al cliente.

4.1.14 Análisis de vulnerabilidades

La seguridad de la red comprende tanto la protección física de los dispositivos como también la integridad, confidencialidad y autenticidad de las transmisiones de datos que circulen por la red.

La seguridad de la red abarca los puntos listados a continuación:

- La información debe estar protegida de una posible destrucción o modificación accidental o intencional (integridad de la información).
- Los datos no deben estar disponibles a los accesos no autorizados (privacidad).
- Las transacciones no deben ser modificadas en su trayecto y se debe asegurar que la entidad que generó la comunicación es quien dice ser. (integridad, autenticidad y no repudio).
- Se debe mantener la integridad física de los dispositivos de red como servidores, switches, etc.
- La red debe estar disponible permanentemente y trabajar con la eficiencia necesaria aún ante fallas inesperadas. (disponibilidad).

El análisis de vulnerabilidades permite comprender la robustez de cada uno de los dispositivos y sistemas ante ataques y obtener información de cuáles son las contramedidas que se pueden aplicar para contrarrestar estos posibles ataques.

El análisis de vulnerabilidades debe ser realizado periódicamente o cuando ocurran cambios importantes en la red, por ejemplo cambios en el diseño de la red o sistemas, actualizaciones de dispositivos.

4.1.14.1 Métodos de análisis de vulnerabilidades

4.1.14.1.1 Caja negra

Se proporciona al analista sólo información de acceso a la red o al sistema. A partir de esta información, el analista debe obtener toda la información posible.

4.1.14.1.2 Caja blanca

El analista de seguridad tiene una visión total de la red a analizar, así como acceso a todos los equipos con permisos de súper usuario o administrador. Este tipo de análisis es más completo.

4.1.14.1.3 Test de penetración

Es un procedimiento en el cual el analista simula ser un atacante y realiza intentos de ataques con el objetivo de encontrar vulnerabilidades. Los resultados encontrados en el test de penetración servirán para obtener una idea general del estado de la seguridad del sistema frente a posibles ataques. El resultado final del

análisis conduce a la obtención del listado de vulnerabilidades, las contramedidas a tomarse y las recomendaciones finales hechas por el analista de seguridad.

4.1.14.2 Tipos de vulnerabilidades

- Vulnerabilidades de implementación
- Vulnerabilidades de configuración
- Vulnerabilidades de dispositivo
- Vulnerabilidades de protocolo
- Vulnerabilidades de aplicación

4.1.14.3 Herramientas para encontrar vulnerabilidades

- Escaneo de puertos
- Analizador de protocolos
- Password crackers
- Ingeniería social
- Trashing

4.2 Implementación de la seguridad en la tienda virtual

4.2.1 OpenSSL

Para la implementación de la seguridad en la tienda virtual vamos a utilizar el programa OpenSSL[12] que consiste en un robusto paquete de herramientas de Criptografía.

Para instalar OpenSSL en Ubuntu Linux se utiliza el siguiente comando: (En otras distribuciones se utiliza comandos similares.)

roberto@ubuntu:~$ ***sudo apt-get install openssl***

Con este comando GNU Linux empieza a descargar el paquete y una vez finalizada la descarga prosigue con su instalación.

[12] (Cox, y otros s.f.)

4.2.1.1 Algoritmos de encriptación disponibles en OpenSSL

Con el siguiente comando podemos obtener el listado de los algoritmos de encriptación disponibles en nuestra versión de OpenSSl.

```
roberto@ubuntu:~$ openssl ciphers -v
DHE-RSA-AES256-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
DHE-DSS-AES256-SHA      SSLv3 Kx=DH       Au=DSS  Enc=AES(256)  Mac=SHA1
AES256-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
EDH-RSA-DES-CBC3-SHA    SSLv3 Kx=DH       Au=RSA  Enc=3DES(168)
Mac=SHA1
EDH-DSS-DES-CBC3-SHA    SSLv3 Kx=DH       Au=DSS  Enc=3DES(168)
Mac=SHA1
DES-CBC3-SHA            SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
DES-CBC3-MD5            SSLv2 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=MD5
DHE-RSA-AES128-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
DHE-DSS-AES128-SHA      SSLv3 Kx=DH       Au=DSS  Enc=AES(128)  Mac=SHA1
AES128-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
RC2-CBC-MD5             SSLv2 Kx=RSA      Au=RSA  Enc=RC2(128)  Mac=MD5
RC4-SHA                 SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=SHA1
RC4-MD5                 SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
RC4-MD5                 SSLv2 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
EDH-RSA-DES-CBC-SHA     SSLv3 Kx=DH       Au=RSA  Enc=DES(56)
Mac=SHA1
EDH-DSS-DES-CBC-SHA     SSLv3 Kx=DH       Au=DSS  Enc=DES(56)
Mac=SHA1
DES-CBC-SHA             SSLv3 Kx=RSA      Au=RSA  Enc=DES(56)   Mac=SHA1
DES-CBC-MD5             SSLv2 Kx=RSA      Au=RSA  Enc=DES(56)   Mac=MD5
EXP-EDH-RSA-DES-CBC-SHA SSLv3 Kx=DH(512)  Au=RSA  Enc=DES(40)
Mac=SHA1 export
EXP-EDH-DSS-DES-CBC-SHA SSLv3 Kx=DH(512)  Au=DSS  Enc=DES(40)
Mac=SHA1 export
EXP-DES-CBC-SHA         SSLv3 Kx=RSA(512) Au=RSA  Enc=DES(40)
Mac=SHA1 export
EXP-RC2-CBC-MD5         SSLv3 Kx=RSA(512) Au=RSA  Enc=RC2(40)   Mac=MD5
export
EXP-RC2-CBC-MD5         SSLv2 Kx=RSA(512) Au=RSA  Enc=RC2(40)   Mac=MD5
export
EXP-RC4-MD5             SSLv3 Kx=RSA(512) Au=RSA  Enc=RC4(40)   Mac=MD5
export
EXP-RC4-MD5             SSLv2 Kx=RSA(512) Au=RSA  Enc=RC4(40)   Mac=MD5
export
roberto@ubuntu:~$
```

4.2.1.2 Certificado digital autofirmado

Con el siguiente comando podemos obtener un certificado digital auto firmado. Especificamos el número de días de validez con el parámetro -days, utilizamos el algoritmo RSA y una clave de 1024 bits. Con el parámetro -keyout definimos que se genere un par de claves pública y privada y que se almacenan en el mismo archivo mycert.pem. Posteriormente la consola nos pide ingresar nuestra información como código de país, estado, ciudad, nombre de la unidad y organización a la que pertenecemos.

roberto@ubuntu:/tmp/openssl$ ***openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem***
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'mycert.pem'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:EC
State or Province Name (full name) [Some-State]:Loja
Locality Name (eg, city) []:Loja
Organization Name (eg, company) [Internet Widgits Pty Ltd]:JG Exclusividades
Organizational Unit Name (eg, section) []:Sw development
Common Name (eg, YOUR name) []:www.jgexclusividades.com
Email Address []:roberto.jacome@gmail.com
roberto@ubuntu:/tmp/openssl$

Para comprobar que nuestro recién creado certificado funciona correctamente podemos utiliza el siguiente comando:

```
roberto@ubuntu:/tmp/openssl$ openssl x509 -text -in mycert.pem
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number:
         85:2c:e1:5e:65:7d:7d:d4
      Signature Algorithm: sha1WithRSAEncryption
      Issuer: C=EC, ST=Loja, L=Loja, O=JG Exclusividades, OU=Sw development,
CN=www.jgexclusividades.com/emailAddress=roberto.jacome@gmail.com
      Validity
         Not Before: Jun 29 13:34:32 2010 GMT
         Not After : Jun 29 13:34:32 2011 GMT
      Subject: C=EC, ST=Loja, L=Loja, O=JG Exclusividades, OU=Sw development,
CN=www.jgexclusividades.com/emailAddress=roberto.jacome@gmail.com
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
            Modulus (1024 bit):
               00:da:c6:c2:94:d3:b7:20:df:bf:d8:f1:69:35:f9:
               f9:32:96:48:48:ca:f9:fa:85:cc:e5:ac:98:d6:31:
               5f:7d:b2:f5:84:d7:20:59:a3:3c:70:f2:ce:c0:18:
               8c:ba:f3:f8:d6:23:34:db:97:d2:f6:7c:7d:05:4f:
               cd:f8:16:df:6a:bc:87:02:99:d1:e9:15:76:28:5a:
               d3:0b:9b:71:d4:fa:5f:38:af:1b:90:ca:44:64:23:
               a3:75:31:be:dc:81:78:39:f1:a6:19:3f:2f:8a:ca:
               15:a4:e7:29:fd:f7:05:79:39:3e:5b:91:86:42:24:
               05:00:30:05:c1:17:69:bf:a5
            Exponent: 65537 (0x10001)
      X509v3 extensions:
         X509v3 Subject Key Identifier:
            D7:53:3C:F8:8E:72:72:B2:BC:00:1E:79:72:10:B0:6B:9D:F0:4F:63
         X509v3 Authority Key Identifier:
            keyid:D7:53:3C:F8:8E:72:72:B2:BC:00:1E:79:72:10:B0:6B:9D:F0:4F:63
            DirName:/C=EC/ST=Loja/L=Loja/O=JG Exclusividades/OU=Sw
development/CN=www.jgexclusividades.com/emailAddress=roberto.jacome@gmail.c
om
            serial:85:2C:E1:5E:65:7D:7D:D4

         X509v3 Basic Constraints:
            CA:TRUE
```

```
    Signature Algorithm: sha1WithRSAEncryption
         a9:dd:08:c6:d1:1f:3e:d7:fa:58:cc:a1:87:68:e0:fc:b4:0f:
         66:2b:be:3c:06:9b:b4:a3:c4:fc:be:4a:86:a8:96:8b:d6:7a:
         9e:70:30:58:db:6c:b2:c9:b4:a4:74:63:c4:ba:46:0b:9b:f3:
         f0:b3:da:e6:e6:8a:85:d2:96:95:32:00:6e:c6:31:d5:12:21:
         5b:5d:99:47:d2:35:2f:c2:b3:f9:52:da:27:6a:58:66:f4:2d:
         17:4d:81:89:1d:05:fb:68:e1:a3:c5:61:95:70:ef:39:b7:93:
         d7:71:31:0d:ad:a2:7a:a8:d0:ab:a6:e0:00:62:b1:e0:b7:4d:
         2b:ce
-----BEGIN CERTIFICATE-----
MIID7zCCA1igAwIBAgIJAIUs4V5lfX3UMA0GCSqGSIb3DQEBBQUAMIGsMQswCQ
YD
VQQGEwJFQzENMAsGA1UECBMETG9qYTENMAsGA1UEBxMETG9qYTEaMBgG
A1UEChMR
SkcgRXhjbHVzaXZpZGFkZXMxFzAVBgNVBAsTDlN3IGRldmVsb3BtZW50MSEwHw
YD
VQQDExh3d3cuamdleGNsdXNpdmlkYWRlcy5jb20xJzAlBgkqhkiG9w0BCQEWGHJv
YmVydG8uamFjb21lQGdtYWlsLmNvbTAeFw0xMDA2MjkxMzM0MzJaFw0xMTA2Mjkx
MzM0MzJaMIGsMQswCQYDVQQGEwJFQzENMAsGA1UECBMETG9qYTENMAsGA1UEBxME
TG9qYTEaMBgGA1UEChMRSkcgRXhjbHVzaXZpZGFkZXMxFzAVBgNVBAsTDlN3IGRl
dmVsb3BtZW50MSEwHwYDVQQDExh3d3cuamdleGNsdXNpdmlkYWRlcy5jb20xJzAl
BgkqhkiG9w0BCQEWGHJvYmVydG8uamFjb21lQGdtYWlsLmNvbTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEA2sbClNO3IN+/2PFpNfn5MpZISMr5+oXM5ayY1jFf
fbL1hNcgWaM8cPLOwBiMuvP41iM025fS9nx9BU/N+BbfaryHApnR6RV2KFrTC5tx
1PpfOK8bkMpEZCOjdTG+3IF4OfGmGT8visoVpOcp/fcFeTk+W5GGQiQFADAFwRdp
v6UCAwEAAaOCARUwggERMB0GA1UdDgQWBBTXUzz4jnJysrwAHnlyELBrnfBPY
zCB
4QYDVR0jBIHZMIHWgBTXUzz4jnJysrwAHnlyELBrnfBPY6GBsqSBrzCBrDELMAkG
A1UEBhMCRUMxDTALBgNVBAgTBExvamExDTALBgNVBAcTBExvamExGjAYBgN
VBAoT
EUpHIEV4Y2x1c2l2aWRhZGVzMRcwFQYDVQQLEw5TdyBkZXZlbG9wbWVudDEh
MB8G
A1UEAxMYd3d3LmpnZXhjbHVzaXZpZGFkZXMuY29tMScwJQYJKoZIhvcNAQkBFh
hy
b2JlcnRvLmphY29tZUBnbWFpbC5jb22CCQCFLOFeZX191DAMBgNVHRMEBTADA
QH/
MA0GCSqGSIb3DQEBBQUAA4GBAKndCMbRHz7X+ljMoYdo4Py0D2YrvjwGm7Sjx
Py+
```

```
SoaolovWep5wMFjbbLLJtKR0Y8S6Rgub8/Cz2ubmioXSlpUyAG7GMdUSIVtdmUfS
NS/Cs/lS2idqWGb0LRdNgYkdBfto4aPFYZVw7zm3k9dxMQ2tonqo0Kum4ABiseC3
TSvO
-----END CERTIFICATE-----
roberto@ubuntu:/tmp/openssl$
```

Con lo que podemos ver la información del certificado, la clave pública generada y todos los datos de nuestra organización que fueron pedidos.

Otra forma de verificación es utilizar el comando:

```
roberto@ubuntu:/tmp/openssl$ openssl verify mycert.pem
mycert.pem: /C=EC/ST=Loja/L=Loja/O=JG Exclusividades/OU=Sw
development/CN=www.jgexclusividades.com/emailAddress=roberto.jacome@gmail.c
om
error 18 at 0 depth lookup:self signed certificate
OK
roberto@ubuntu:/tmp/openssl$
```

Con lo que obtenemos los datos del certificado y el mensaje OK. Sin embargo obtenemos la advertencia **self signed certificate ,** que quiere decir que el certificado es auto firmado, en otras palabras que no ha sido verificado por una Autoridad de Certificación y por lo tanto no es recomendable para aplicaciones de Comercio Electrónico.

4.2.1.3 Generación de par de claves pública y privada

Pedemos crear una clave privada con el algoritmo RSA mediante el siguiente comando y ver el resultado:

```
roberto@ubuntu:/tmp/openssl$ openssl genrsa -out miclave.pem 1024
Generating RSA private key, 1024 bit long modulus
.........................++++++
.............++++++
e is 65537 (0x10001)
roberto@ubuntu:/tmp/openssl$
```

La clave generada se guarda en el fichero miclave.pem

Si queremos ver el contenido de este archivo podemos utilizar el comando **less miclave.pem** y poder apreciar el aspecto de nuestra clave privada:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCw9J5200sAoHCLah3Azfore8ytq84MMBV3pUEFnfQJpQk1YM97
9Nmm/O5TJPIZM88i4yMQMPDAR4Y32TJDJw3NvwFH/Dz5LG4c1vSuoVDNsoZdD3
iw
TfMNqjAmv3o0atKp0b3rqOoVEJya9FQPB93WPFX4lHvjxEk2GOhgFi3qFQIDAQAB
AoGAUr0d9utWr2VVGiq03Liyl3BcRHwHbLVy5C2VmH3BZF0RUC8C9K+IsUakXKX
y
uQv7dxGDDIgFmbIXtLdOhA2Xqurm1JcHGnS7MXjB2dDDXrdNyGGZKACwuHUPXp
Li
lV4AQNvUWc2nALlj9JTdiREm16YtVfAiFdO6F7zs6f9m28ECQQDrxWmHMOozhOK
E
Cuaa67O3jXd8HroeL6TKqYassmf0dbH5RGCoGZhFOqf7i+jhoCLLVcmPNfDGgiHa
7GYwgQdRAkEAwCNaekNCbXss37jHXB/7ikTZHb5qslFfYnBUrqlahZrB7Sqj0t4V
yHHQaWNa83TGI1sxtO4USwN6AWB+0+YNhQJAdoGSqlYm0eBwLqiAAsyIV7Fupl+
X
c0bFUOxm+fTulYx2+XCqGLRMujdiaeiIFKh19LhZCHe3Xz/Xnn+GWcjxEQJBALYm
TfYuoWENLwOBN25824i/sF1kUd5vClBWUdGrt6eFiWd/zwNvi5MVYxEmIek/wT9v
ivYN5qsexqFbi6DzLckCQQC91orWbJ6E3OvVC7JtRr+E+IOm8PwRGDEIV8hhvhGb
4o1pw1vSOxeRFckW068ljg/ApjuczXLAG9WaNvjKs+Ly
-----END RSA PRIVATE KEY-----
miclave.pem (END)
```

A partir de la clave privada podemos generar la clave pública con el siguiente comando y ver el resultado:

```
roberto@ubuntu:/tmp/openssl$ openssl rsa -in miclave.pem -pubout
writing RSA key
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCw9J5200sAoHCLah3Azfore
8yt
q84MMBV3pUEFnfQJpQk1YM979Nmm/O5TJPIZM88i4yMQMPDAR4Y32TJDJw3Nv
wFH
/Dz5LG4c1vSuoVDNsoZdD3iwTfMNqjAmv3o0atKp0b3rqOoVEJya9FQPB93WPFX4
lHvjxEk2GOhgFi3qFQIDAQAB
-----END PUBLIC KEY-----
roberto@ubuntu:/tmp/openssl$
```

4.2.1.4 Obtener la función de Hash

Podemos obtener la función de Hash de un archivo utilizando los algoritmos MD5 y SHA1 con los siguientes comandos:

```
roberto@ubuntu:/tmp/openssl$ openssl dgst -md5 confidencial.txt
MD5(confidencial.txt)= e14504df2bc2dd0cbb50dc4bd6784c71

roberto@ubuntu:/tmp/openssl$ openssl dgst -sha1 confidencial.txt
SHA1(confidencial.txt)= c20f7eabee917b23616de16f2cece2ec47bba276
roberto@ubuntu:/tmp/openssl$
```

4.2.1.5 Firmar digitalmente un archivo

Una vez generadas las claves privada y pública es factible firmar digitalmente nuestros propios archivos para garantizar que cuando sean enviados éstos no hayan sido modificados por terceros. Utilizamos el siguiente comando para firmar digitalmente un archivo llamado confidencial.txt con nuestra clave privada almacenada en el archivo miclave.pem

```
roberto@ubuntu:/tmp/openssl$ openssl dgst -sha1 -sign miclave.pem -out
confidencial.txt.sha1 confidencial.txt
roberto@ubuntu:/tmp/openssl$
```

Este comando nos genera un archivo binario el mismo que tiene nuestro documento firmado digitalmente.

Para verificar la firma digital utilizamos nuestra clave pública generada almacenada en el archivo clavepub.pem, el archivo binario y el archivo original:

```
roberto@ubuntu:/tmp/openssl$ openssl dgst -sha1 -verify clavepub.pem -
signature confidencial.txt.sha1 confidencial.txt
Verified OK
roberto@ubuntu:/tmp/openssl$

roberto@ubuntu:/tmp/openssl$ openssl enc -aes-256-cbc -salt -in confidencial.txt -
out confidencial.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
roberto@ubuntu:/tmp/openssl$ openssl enc -d -aes-256-cbc -in confidencial.enc
enter aes-256-cbc decryption password:
contenido
este es un mensaje secreto
roberto@ubuntu:/tmp/openssl$
```

4.2.2 Definición de las Políticas de seguridad de la tienda virtual

- Se definirán 2 tipos de usuarios de la tienda virtual. Los usuarios que son empleado de la empresa y realizan tareas de operación y administración de la tienda virtual y los clientes de la tienda virtual.
- Cada usuario de la tienda virtual deberá tener una cuenta única y será responsable del correcto uso de la misma.
- Todo usuario (operador o administrador) deberá firmar un compromiso de seguridad y confidencialidad del uso del sistema. Este compromiso debe ser renovado cada año.
- Todo usuario deberá respetar y no divulgar la información de la empresa que sea considerada relevante para los intereses de la misma.
- Cada usuario es responsable de los sitios y la información que accede con su cuenta.
- Se deberá dar la oportuna capacitación a los nuevos usuarios (operadores) de la tienda virtual.
- Las contraseñas para un operador / digitador de la tienda virtual deben tener como mínimo 8 caracteres.

- Las contraseñas deben tener un periodo de caducidad de 3 meses.
- Las contraseñas deben ser personales y no deben ser divulgadas bajo ningún concepto.
- El administrador de la base de datos deberá revisar los intentos de acceso no autorizados a la base de datos con el objetivo de prevenir pérdida de información valiosa.
- Se deberá respaldar la información de la tienda virtual diariamente.
- Las contraseñas utilizadas deberán ser almacenadas en forma cifrada en la base de datos de la tienda virtual.
- En caso de que un empleado renuncie a la empresa, su cuenta de usuario debe ser inmediatamente deshabilitada.
- Se deberá limitar el acceso a los operadores de la tienda virtual a la información sensible de la empresa.
- Se deberá utilizar programas antivirus para proteger el sistema y la base de datos de todo tipo de amenazas.
- Se deberá utilizar certificados digitales firmados por una autoridad de certificación para proteger la información enviada por los clientes.
- Se conducirán pruebas periódicas del correcto funcionamiento de las funcionalidades de la tienda virtual.
- El administrador de la base de datos tendrá privilegios especiales con los que podrá tener acceso a todas las partes del sistema.
- En caso de cambiar de administrador de la base de datos las contraseñas utilizadas deberán ser cambiadas de forma inmediata.

Capítulo 5

Open Source

5.1 Introducción teórica

5.1.1 Open Source

Open Source[13] está compuesto por software en el que el código fuente es publicado para su libre utilización, modificación y distribución. El desarrollo del Internet y el free software contribuyeron enormemente al crecimiento del movimiento Open Source.

En el año de 1969 la Agencia de Proyectos de Investigación Avanzada (Advanced Research Projects Agency ARPA) del Departamento de Defensa de los Estados Unidos, presentó la red de computadoras llamada ARPAnet (Advanced Research Projects Agency Network).

ARPAnet empezó a ser utilizada para investigación y por el sector privado y esto llevó al desarrollo de Internet. Mediante el intercambio de código fuente a través de esta red se llevaron a cabo actividades para mejoramiento del software, lo que fue basado en la idea principal de free software (ética de hacker), que establece que "todos deben estar permitidos de obtener y modificar el código fuente para desarrollar mejores programas". Esta idea contribuyó al desarrollo de Internet y también a la idea de software libre.

El concepto de software libre fue propuesto originalmente por Richard Stallman en el MIT. Stallman lanzó Emacs un editor y GCC un compilador como productos de software libre y fundó FSF (Free Software Foundation) estableciendo la idea de software libre como GNU (GNU is Not Unix).

La idea de software libre fue muy rígida en su intento de asegurar la libertad, y ciertos negocios y usuarios no podían utilizar esta idea debido a su rigidez a pesar de que querían utilizarla. Debido a que estos negocios no podían utilizar la idea de software libre, esto se constituyó en un obstáculo para la difusión de este tipo de software. Eric Raymond quien trabajó en la difusión y desarrollo de software libre conjuntamente con Stallman, definió una técnica para romper la barrera del uso del software libre y utilizaron el Internet para formar grupos de desarrolladores voluntarios conocidos como "comunidades", los mismos que querían desarrollar

[13] (Open Source Community 2009)

software que les gustaba. “Open Source” fue creado para definir los aspectos no clarificados y proponer nuevos conceptos.

5.1.1.1 Definición de Open Source

Los siguientes aspectos tienen que ver con la definición de Open Source:

1) Redistribución libre.

   La licencia no debe restringir a ninguna parte para vender o dar el software ya sea como un componente o una mejora. La licencia no requerirá ningún pago por esta venta.

2) Código fuente

   El programa debe incluir el código fuente, y debe permitir la distribución del código fuente como del programa compilado. En caso de que un producto no sea distribuido con el código fuente, deberá existir un medio bien publicitado de obtener el código fuente, por ejemplo descargarlo de Internet sin ningún costo.

3) Trabajos derivados

   La licencia debe permitir modificaciones y trabajos derivados y debe permitir que estos trabajos sean distribuidos bajo los mismos términos que constan en la licencia del software original.

4) Integridad del código fuente del autor

   La licencia puede restringir el código fuente de ser distribuido con modificaciones. La licencia puede requerir que trabajos derivados lleven un nombre diferente o un número de versión diferente del software original.

5) No discriminación en contra de personas o grupos

   La licencia no debe discriminar contra ninguna persona o grupo de personas.

6) No discriminación en contra de algún campo o empresa

   La licencia no debe restringir a nadie del uso del programa en un campo o emprendimiento específico.

7) Distribución de la licencia

Los derechos incluidos en el programa deben aplicar a todos aquellos a los que el programa ha sido redistribuido sin ninguna licencia adicional para ellos.

8) La licencia no debe ser específica de un producto.

   Los derechos incluidos en el programa no deben depender de alguna distribución del programa.

9) La licencia no debe restringir otro software.

   La licencia no debe imponer restricciones sobre otros programas que sean distribuidos con el software licenciado.

10) La licencia debe ser tecnológicamente neutral.

    No debe haber prohibiciones de la licencia sobre ninguna tecnología o estilo de interfaz.

5.1.1.2 Ventajas y desventajas del software Open Source

5.1.1.2.1 Ventajas

Desde el punto de vista del negocio, el software Open Source es sin costo o poco costoso.

Desde el punto de vista técnico se tiene varias ventajas como:

- Los errores pueden ser encontrados revisando el código fuente.
- Los errores pueden ser corregidos.
- Las operaciones y funciones pueden ser probadas.
- El código fuente representa el documento más útil.
- Muy buena documentación está disponible.

5.1.1.2.2 Desventajas

- Se necesitan algunas ideas para proteger los derechos de propiedad intelectual.
- Se tiene cortos intervalos de implementación de funciones.

- Las nuevas versiones no son siempre entregadas como se esperaban.
- Los arreglos a los errores encontrados por usuarios no son siempre reflejados en el código fuente.
- La compatibilidad con versiones anteriores no siempre es garantizada.

5.1.1.3 Razones por las que el software Open Source es desarrollado

5.1.1.3.1 Punto de vista de los desarrolladores

- Ninguno de los programas que ellos quieren están disponibles.
- Es menos pesado y más divertido desarrollar en grupo que individualmente.
- La historia del desarrollo puede ser obtenida y conocida.

5.1.1.3.2 Punto de vista de las empresas

- Efectos de publicidad, contribución a la comunidad.
- Puede utilizar parcialmente el software Open Source y ciertas características tomarlas de otros proveedores.
- Puede reducir la carga y costo de soporte de software

5.1.1.3.3 Punto de vista del Gobierno

- Puede beneficiarse de la experiencia de países que utilizan el mismo tipo de software en todo el mundo.
- Fácil de detectar fallos de seguridad.
- Desarrollo de sistemas contribuye a la revitalización industrial

5.1.1.3.4 Punto de vista del usuario

- Se beneficia de la característica “sin costo” o en algunos casos costo mínimo.
- Se beneficia de que pueden contribuir al desarrollo y dar sugerencias para mejorar las versiones.
- Preocupación porque los desarrolladores de software propietario dejen de desarrollar algún software específico.

5.1.1.4 Listado de sistemas Open Source más importantes

5.1.1.4.1 GNU Linux

Categoría: sistema operativo

En el año de 1991, Linus Torvalds inició el desarrollo de Linux en un Intel 386. A pesar de que Linux fue originalmente el nombre del Kernel, también es usado como un término genérico para las diferentes distribuciones.

Existen varias distribuciones desarrolladas alrededor del mundo siendo las principales: Red Hat, Fedora, Mandriva, SuSe, CentOS, Ubuntu.

GNU Linux utiliza la licencia GPL.

Características:

- El kernel Linux tiene las siguientes funciones: manejo de procesos, operación de interrupciones, procesamiento de Entrada/Salida, manejo del sistema de archivos, manejo de memoria, manejo de red, manejo de dispositivos.
- Provee funciones equivalentes a las familias System V y BSD.

5.1.1.4.2 PostgreSQL

Categoría: Sistema de gestión de base de datos.

Es una base de datos relacional desarrollada en el Departamento de Ciencias de la Computación de la Universidad Barkley de California.

Características:

- Funciona con diferentes plataformas.
- Posee una poderosa función de manejo de transacciones.
- Implementa los estándares de SQL.
- Datos grandes pueden ser procesados.
- Utiliza licencia BSD.
- La implementación puede ser realizada con diferentes APIs.

5.1.1.4.3 MySQL

Categoría: Sistema de Gestión de Base de Datos.

Se originó a partir de UNIREG, una herramienta de administración de base de datos desarrollado por Michael Widenius en 1979. En 1994 UNIREG fue integrado con mSQL y en el año 1995 MySQL 1.0 fue lanzado. Posee dos tipos de licencia propietaria y GPL.

Características:

- Multi hilo, corre bajo casi todas las versiones de sistemas operativos UNIX.
- Varios controles de acceso están disponibles, no tienen fallas de memoria.
- Posee funciones de optimización como reparación, respaldo, replicación de la base de datos.
- La implementación puede ser realizada con diferentes APIs.

5.1.1.4.4 Apache

Categoría: Servidor Web

Desarrollado basado en NCSA httpd1.3 de la NCSA (National Center for Supercomputing Application), un laboratorio para estudio de aplicaciones de súper computación.

El nombre Apache surgió luego al adoptar el nombre de una tribu nativa norteamericana. Originalmente consistía en múltiples programas (parches) utilizados para corregir un número considerable de errores incluidos en NCSA httpd.

Corre con la familia de sistemas operativos de UNIX, Mac y Windows.

La licencia utilizada es la Apache Software License.

Características:

- Es el servidor web más utilizado en Internet.
- El proyecto Apache originó muchos proyectos relacionados.

5.1.1.4.5 Tomcat

Categoría: Servidor Web

Software Open Source desarrollado por el sub proyecto Jakarta Project, que es un servidor de aplicaciones para los Servlets de Java y JSP.

Características

- Apache puede ser usado independientemente como un servidor Web, sin embargo su mayor aplicación tiene que ver con la ejecución de plug-ins que corren con Apache o Internet Information Server.
- Funciona con las familias de sistemas operativos: UNIX, Windows y Max OS X.
- Utiliza la licencia Apache Software License, todos pueden usar, modificar o redistribuir Tomcat sin ningún cobro o restricción.

5.1.1.4.6 PHP

Categoría: Lenguaje de programación para la Web.

PHP es el acrónimo de Hypertext Preprocessor, Es un lenguaje de script utilizado para generar archivos HTML como el JavaScript o ASP (Active Server Pages). En 1995, PHP/FI fue desarrollado por Rasmus Lerdorf

Características:

- Provee funciones que corren en varias bases de datos como Oracle, MySQL, PostgreSQL.
- Debido a que PHP es implementado como parte de módulos en Apache, corre más rápido que el lenguaje PERL y con menos carga al servidor.
- Corre en la familia de sistemas operativos UNIX y Windows.
- Fácil implementación y manejo de sesión.
- Fácil proceso de verificación y corrección de errores (debugging).

5.1.1.4.7 Eclipse

Categoría: Ambiente de desarrollo de software integrado

Software libre de código abierto de tipo IDE (Integrated Development Environment).

Originalmente IBM desarrolló Eclipse para uso interno como ambiente de desarrollo para servidores de aplicaciones. En noviembre de 2001, el código fuente fue donado a la comunidad Open Source.

Actualmente el proyecto Eclipse (http://www.eclipse.org), está a cargo del desarrollo.

Características:

- Arquitectura plug-in
- Rápido rendimiento, excelente ayuda.
- Condiciones de licencia flexibles

5.1.1.4.8 Mozilla

Categoría: Navegador Web

Es un navegador Web de código abierto desarrollado por Netscape Corporation, es el sucesor de Netscape Communicator 4.

Su desarrollo es conducido principalmente por Mozilla.org en el cual muchos de los ingenieros de Netscape están participando. (Mozilla.org también desarrolla y distribuye otros productos de software como BugZilla, un sistema de seguimiento de reportes).

Características

- Al igual que Netscape Communicator, Mozilla fue originalmente desarrollado para correr bajo múltiples plataformas. Por tanto muchas funciones son escritas en código texto excepto para las partes dependientes de la plataforma.

5.2 Utilización de herramientas Open Source en el desarrollo de la tienda virtual

Para el desarrollo, pruebas e implementación de la tienda virtual de e-commerce es necesario contar con paquetes de software de diferentes características, como sistemas operativos, sistemas de gestión de bases de datos, servidores web, entornos de desarrollo, herramientas para modelado de datos, etc. En el presente proyecto se ha tratado de dar mayor relevancia al software de código abierto antes que el software propietario.

Tipo	Herramienta Propietaria	Herramienta Open Source sustituta
Sistema operativo	Microsoft Windows Server	GNU Linux
Servidor Web	Microsoft Internet Information Server	Apache Tomcat
Gestor de base de datos	Oracle Standard Edition	MySQL
Lenguaje de programación	Microsoft .NET	Java J2EE de Sun
Editor HTML	Microsoft Frontpage	Macromedia DreamWeaver, Quanta
Modelado UML	Rational Rose	Dia, Umbrello, ArgoUML
Diagramación y Diseño	Microsoft Visio	Kvisio
Ofimática	Microsoft Office	OpenOffice de Sun
Criptografía		OpenSSL

Tabla 5.1 Herramientas Open Source sustitutas de herramientas propietarias

El sistema operativo para el desarrollo del trabajo es el Sistema Operativo Linux distribución Ubuntu, debido a que el servidor que alojará la tienda virtual estará hospedada en un ambiente Linux. Se ha creído conveniente utilizar este sistema operativo debido a que se puede tener un ambiente muy similar al que la tienda virtual de Comercio Electrónico estará alojada.

5.2.1 GNU / Linux

La gran mayoría de los programas Open Source pueden ser fácilmente conseguidos a través de Internet, y GNU Linux no es la excepción. Para ello debemos ir a la dirección web de la distribución que mejor se ajuste a nuestras necesidades, descargarla y quemar el disco ISO en CD o DVD el mismo que nos permitirá arrancar e instalar GNU Linux.

A continuación se presentan un listado de distribuciones ampliamente utilizadas:

Distribución	**Dirección Web**
CentOS	http://www.centos.org
Red Hat	http://www.redhat.com
Fedora	http://fedoraproject.org
SuSe	http://es.opensuse.org
Mandriva	http://www.mandriva.com
Ubuntu	http://www.ubuntu.com
Debian	http://www.debian.org

Tabla 5.2 Distribuciones GNU Linux

La instalación de paquetes de software y la actualización de los sistemas GNU Linux se la hace fácilmente a través de Internet.

5.2.2 MySQL

MySQL es un sistema de gestión de base de datos Open Source ampliamente extendido y utilizado en aplicaciones Web. Es un paquete de software sencillo pero potente. Para el desarrollo del presente proyecto se ha utilizado la versión 5.1.

Para desarrollar paquetes informáticos con MySQL en ambientes GNU Linux debemos instalar 2 componentes el servidor y el cliente de MySQL. Mediante una conexión a Internet, podemos instalar estos 2 componentes con el siguiente comando en la distribución Ubuntu:

$ sudo get-install mysql-server mysql-client

GNU Linux realizará la descarga de los paquetes indicados, verificará dependencias y empezará con la instalación de los mismos.

Una vez instalado el sistema de gestión de base de datos MySQL, es recomendable crear un usuario en MySQL que será el dueño de la base de datos.

Para crear la base de datos de la tienda virtual utilizamos el siguiente comando

> create database jgexclus_base

Una vez creada la base de datos de la tienda virtual procedemos a ejecutar el script de creación de las tablas. Para ello utilizamos el comando **source**.

> source /home/usuario/scripts/inicio.sql

5.2.3 Dia

En el modelo de desarrollo de software clásico o modelo en cascada, tenemos las 2 primeras etapas que son el análisis y el diseño de la aplicación. Estas etapas son fundamentales para el éxito del proyecto. El equipo de desarrollo de software utiliza los diagramas UML para llevar a cabo tan trascendental tarea, lo que en épocas pasadas los analistas de sistemas utilizaban los diagramas de flujo y otras herramientas de análisis.

El paquete Dia es una herramienta que permite la elaboración de diagramas UML como diagramas de casos de uso, diagramas de secuencia, diagramas de estado, diagramas de clases, el diagrama entidad relación, etc. Dia es un paquete Open Source y realiza las mismas funciones que paquetes comerciales como el Rational Rose.

5.2.4 Apache Tomcat 6.0

El contenedor Web Tomcat es el programa que permitirá el funcionamiento de la tienda virtual de Comercio Electrónico, puesto que es el programa que escucha las peticiones hechas por los clientes (navegadores Web), procesará estos pedidos y retornará la información correspondiente. Una de las grandes ventajas de Tomcat es que incluye los módulos del servidor Web Apache, por lo que también admite páginas hechas en el lenguaje PHP.

Luego de descargar la versión de Tomcat para Linux se procedió a la instalación sin embargo se presentaron algunos problemas al momento de probar el contenedor web

Al buscar en foros especializados de internet se halló que el puerto está ocupado por otro proceso y ese es el motivo por el cual no funciona el contenedor web Tomcat, la solución fue configurar un nuevo puerto para que funcione el Tomcat, esta configuración se la realiza en el archivo **$CATALINA_BASE/conf/server.xml**

$CATALINA_BASE es una variable de entorno que indica el directorio de instalación de Tomcat.

5.2.5 NetBeans

Netbeans es un entorno de desarrollo Open Source patrocinado por Sun

http://www.sun.com

Ofrece una gran cantidad de ventajas a los programadores, permite la creación de aplicaciones para entornos cliente – servidor, aplicaciones Web, dispositivos móviles.

El uso de un entorno de desarrollo como NetBeans facilita mucho la programación debido a que automatiza tareas complejas.

5.2.6 Mozilla Firefox

Para realizar las pruebas del proyecto se utilizó el navegador Web Mozilla Firefox, el mismo que es uno de los navegadores más confiables y seguros. Firefox es un paquete Open Source y supera ampliamente a Internet Explorer de Microsoft.

Mozilla Firefox puede ser descargado desde su web

http://www.firefox.org

5.2.7 Quanta

Para la construcción de una aplicación Web es necesario el conocimiento del lenguaje HTML para la creación y estructuración de las páginas de nuestra aplicación. Quanta es un paquete de software Open Source que facilita enormemente esta tarea, realiza las mismas funciones que el programa Microsoft Frontpage puesto que permite crear documentos HTML en un ambiente WYSIWYG (What You See Is What You Get).

5.2.8 OpenOffice

Es la herramienta ofimática más extendida en el mundo Open Source.

Esta herramienta viene incluida con el paquete de instalación de la mayoría de las distribuciones GNU Linux y también se la puede conseguir en su sitio Web:

http:// www.openoffice.org

Su entorno es muy similar al de Microsoft Office y tiene la ventaja de que puede leer desde archivos de Microsoft Office, incluso de la versión 2007, así mismo, permite exportar los documentos a formato PDF.

El paquete de Open Office está compuesto por los siguientes programas:

Nombre	**Tipo de Aplicación**
Writer	Procesador de texto
Calc	Hoja de cálculo
Impress	Creación de diapositivas
Base	Base da datos
Draw	Creación de gráficos
Math	Edición de fórmulas matemáticas

Tabla 5.3 Paquete OpenOffice

5.2.9 OpenSSL

OpenSSL es una poderosa herramienta Open Source para Criptografía. Viene incluido en la mayoría de distribuciones GNU Linux, también es posible descargarla desde su sitio Web:

www.openssl.org

OpenSSL permite realizar tareas de Criptografía como creación de pares de claves pública y privada, creación de certificados digitales, encriptación de archivos, firmar digitalmente archivos, etc. Posee las funciones de una gran cantidad de algoritmos de cifrado lo que lo convierte en una poderosa herramienta.

Capítulo 6

Tendencias futuras

6.1 Mobile Commerce

El m-commerce es la compra y venta de productos o servicios a través de dispositivos de mano conectados de forma inalámbrica como teléfonos celulares o PDAs. La mayor ventaja del comercio móvil es que le permite al usuario acceder a la compra de productos o servicios sin importar la ubicación en la que se encuentre, así mismo es destacable la enorme penetración que tienen los dispositivos móviles en la población.

Con la introducción de tecnologías avanzadas de telefonía celular, los consumidores de dispositivos móviles serán capaces de acceder a contenidos y servicios desde cualquier lugar y a cualquier hora. Por ejemplo, serán capaces de utilizar dispositivos móviles para acceder a sus cuentas bancarias y pagar facturas, leer estados de cuenta, iniciar transacciones de compra/venta, recibir promociones especiales, generar pedidos, etc.

6.1.1 Productos y servicios disponibles a través de los teléfonos móviles

6.1.1.1 Mobile ticketing

Los tickets pueden ser enviados hacia teléfonos móviles usando una variedad de tecnologías, y posteriormente los consumidores pueden utilizar inmediatamente estos tickets mostrándolos por medio de su teléfono en las entradas y puestos de control.[14]

Así mismo, los tickets pueden ser reservados y cancelados en el móvil con la ayuda de aplicaciones simples las mismas que pueden ser descargadas o accedidas a través de portales WAP.

El uso de la tecnología de expendio de tickets a través del teléfono móvil en aeropuertos, estadios, estaciones de tren, permitirá disminuir las congestiones encontradas en las grandes ciudades.

[14] (Widegren 2006)

6.1.1.2 Vouchers y cupones móviles

La tecnología de tickets móviles también puede ser usada para la distribución de vouchers y cupones. El voucher o cupón es representado por un token virtual que es enviado al teléfono móvil. Al presentar el teléfono móvil con uno de estos tokens en el punto de venta asociado, le permite al consumidor recibir los beneficios que ofrece el negocio para los clientes frecuentes.

6.1.1.3 Compra y entrega de contenido

Actualmente, la compra y entrega de contenido móvil consiste en la venta de ringtones, wallpapers y juegos para teléfonos móviles. La convergencia de teléfonos móviles, reproductores de mp3 y reproductores de video en un sólo dispositivo, conducirá al incremento de la compra y entrega de piezas musicales y video, todo esto gracias al aumento de las velocidades de descarga.

6.1.1.4 Servicios basados en la ubicación

A diferencia de las PC de casa, la ubicación de un teléfono móvil constituye en una importante pieza de información usada en las transacciones de comercio móvil. Al conocer la ubicación del consumidor, se puede ofrecer servicios como:

- Las ofertas locales.
- El clima local.
- Seguimiento y monitoreo de personas.

6.1.1.5 Servicios de Información

Una gran variedad de servicios de información puede ser entregada a los usuarios de teléfonos móviles de la misma forma que es entregada a los PCs. Estos servicios incluyen:

- Servicios de noticias.
- Servicio de la bolsa de valores e información financiera
- Resultados deportivos
- Información del tráfico y congestión vehicular

6.1.1.6 Banca móvil

Los bancos y otras entidades financieras están incursionando en el comercio móvil para permitir a sus clientes no sólo el acceso a la información financiera del cliente,

sino también a la posibilidad de realizar transacciones, este servicio es conocido como Mobile Banking o m-banking.

6.1.1.7 Subastas Móviles

En los últimos años las subastas reversas a través del teléfono móvil han ganado popularidad. En contraste con las subastas tradicionales, las subastas a través del móvil cobran al consumidor cada vez que hacen una oferta. Existen operadoras que cobran por suscripción.

6.1.1.8 Navegación Móvil

La navegación móvil permite a los consumidores comprar en línea en cualquier momento y en cualquier lugar.

Existen diversos navegadores para teléfonos móviles como Opera Mini que se destaca por su velocidad de renderización de imágenes y facilidad de uso.

6.1.1.9 Compras Móviles

En lugar de utilizar catálogos impresos, los minoristas pueden enviar a sus clientes una lista de productos a través del teléfono móvil o pudieran visitar la versión móvil del sitio del comerciante. Adicionalmente, los minoristas también serían capaces de hacer un seguimiento de sus clientes y notificar sobre descuentos en sus tiendas físicas.

6.1.1.10 Marketing y publicidad móvil

El Marketing móvil es un concepto naciente, pero su crecimiento es vertiginoso. Tiene que ver con campañas de marketing especialmente de grandes empresas. Se utiliza el m-commerce para la expansión desde servicios hasta publicidad. Sin embargo existen pocas leyes que protejan al consumidor en casos de abusos o publicidad no deseada, sin embargo en los próximos años, será necesario regular este tipo de comunicación, así mismo, la seguridad de la información deberá ser mejorada en los dispositivos móviles.

6.1.2 Métodos de pago a través del móvil[15]

El pago a través del móvil es una alternativa de pago de rápido crecimiento, especialmente en Europa y Asia. En lugar de pagar en efectivo, cheques o tarjetas de crédito, el consumidor puede utilizar su teléfono móvil para pagar una gran variedad de servicios y productos tanto físicos como digitales, por ejemplo:

[15] (Symteco s.f.)

- Música, vídeos, ringtones, suscripciones en línea, wallpapers, y otros servicios digitales.
- Tickets de transporte: buses, metro, trenes; servicios de parqueo.
- Libros, revistas, entradas, etc.

6.1.2.1 Servicio SMS especial

El consumidor envía una petición de pago a través de un mensaje de texto SMS hacia un short code (número especiales cortos) y el recargo es aplicado a la cuenta del consumidor. El comerciante dueño del servicio es informado del éxito de la transacción de pago y posteriormente realiza el envío de los bienes (digitales en la mayoría de los casos).

Las razones por las que este tipo de método de pago no es ampliamente adoptado y se prevé que será reemplazada son las siguientes:

1. Pobre confiabilidad: pagos por transacciones pueden fallar fácilmente si el mensaje se pierde.

2. Velocidad lenta: El envío de mensajes puede ser lento y puede llegar a tomar horas antes de que el comerciante reciba la confirmación del pago, mientras que el consumidor no querrá esperar más de unos pocos segundos.

3. Seguridad: La encriptación SMS/USSD termina en una interface de radio por lo que culmina siendo un mensaje de texto plano.

4. Costo elevado: Se involucran costos de configurar los números especiales (short code), el costo de envío del contenido digital (Multimedia Messaging Service) y los costos de envío de los mensajes por parte del consumidor.

6.1.2.2 Cargo Directo al móvil

El consumidor usa la opción de cargo directo al móvil durante el pago en un sitio de Comercio Electrónico. La autenticación se realiza con un identificador de usuario y una contraseña, la cuenta del móvil es la que recibe el cargo de la compra. La ventaja es que no se requiere una tarjeta de crédito. Este tipo de pago es muy popular en el Asia y presenta las siguientes ventajas:

1. Seguridad: La autenticación y el motor de manejo de riesgos previene los fraudes.
2. Comodidad: No se necesita un registro previo y no se necesita un software adicional para el móvil.
3. Facilidad de uso.

4. Velocidad de proceso.
5. Ampliamente utilizado. Contenido digital es vendido en línea y utiliza este modelo de pago especialmente en Asia.

6.1.2.3 Pagos Web a través del Móvil (WAP)

El consumidor utiliza su navegador Web incorporado, utiliza el protocolo WAP (Wireless Application Protocol) como la tecnología base. Este método presenta la ventaja de la familiaridad con pagos hechos en Internet.

Sin embargo, es necesario el uso de tarjetas de crédito para este tipo de mecanismo de pago.

6.1.2.3 Sin contacto NFC (Near Field Communication)

NFC es un protocolo basado en una interfaz inalámbrica, en donde la comunicación se realiza entre dos entidades (peer to peer). Los pagos mediante NFC son posibles gracias a la infraestructura de las tarjetas inteligentes. El dispositivo del usuario puede realizar transacciones tan solo con tocar el teléfono móvil con un lector de tarjetas NFC o una pasarela de tickets, la información del pago como tarjetas de crédito o pasajes en medios de transporte son grabados de forma segura en la tarjeta inteligente integrada en el móvil. Este tipo de pagos es muy prometedor y puede ser muy conveniente y seguro.

6.1.3 Pagos a través del móvil en Japón

Los usuarios de teléfonos móviles en Japón usan sus dispositivos para comprar en malls virtuales y boutiques en línea[16]. Son capaces de utilizar sus dispositivos para casi todo tipo de pagos y están muy avanzados en comparación con el resto del mundo en materia de pagos mediante el móvil. Las razones principales de tales adelantos tienen que ver con la constante innovación y la predisposición para la inversión en este campo.

NTT DoCoMo, el operador de telefonía móvil más grande del Japón, lanzó dispositivos móviles equipados con chips FeliCa desarrollados por Sony, los mismos que hacían posible que los teléfonos móviles almacenen múltiples formas de datos incluyendo identificación personal, números de cuenta bancaria y saldos, información del crédito bancario, boletos de transporte urbano, etc. Como resultado, además de facilitar los pagos remotos, los móviles de NTT DoCoMo posibilitan al consumidor el uso de sus dispositivos como sustituto del dinero en efectivo y de las

[16] (Darin s.f.)

tarjetas de crédito en las máquinas expendedoras y en almacenes donde está disponible el servicio.

6.2 Realidad virtual

Es posible utilizar tecnologías muy avanzadas como es el caso de la Realidad Virtual en estrategias de Comercio Electrónico[17]. Empresas publicitarias empiezan a ofrecer tours virtuales para observar los productos, navegar por las habitaciones de un hotel, o tener la posibilidad de observar los acabados de un vehículo con todos los detalles, la gran ventaja de la Realidad Virtual es que representa lo más cercano a una visita a una tienda virtual.

Otro tipo de desarrollo es el probador virtual para la venta de ropa por Internet, en la que ya existen prototipos.

6.2.1 Marketing virtual

No existe duda de que el Marketing virtual se convertirá en una industria creciente en las décadas venideras. El marketing virtual permitirá una serie de servicios, tal como alguna vez la televisión condujo a miles de producciones creativas, de la misma manera la tecnología de la realidad virtual será la semilla de alianzas de negocios. Algunos factores a considerar en el desarrollo del Marketing virtual son:

1. Disminución de costos:

   En los actuales momentos los costos de los sistemas de realidad virtual son muy onerosos, sin embargo se espera una importante disminución de estos precios en los próximos años debido al desarrollo de la tecnología y una creciente demanda.

2. El mejoramiento de los sistemas computaciones tanto de hardware como de software, permitirán que los consumidores tengan más opciones en cuanto a sistemas de realidad virtual con un evidente mejoramiento de calidad.
3. Un mayor número de aplicaciones influirá en el uso del marketing virtual, debido a que el marketing virtual es una industria orientada a las aplicaciones y no una industria orientada a la tecnología. Esta característica la comparte con la televisión y las computadoras, por ejemplo, una mayor calidad y cantidad de programas de entretenimiento condujo a mayores ventas de televisores, lo mismo sucedió con los computadores personales, que fueron adquiridos mayormente cuando existían aplicaciones productivas como el procesador de palabras o la hoja de cálculo.

[17] (Wikipedia s.f.)

Por lo tanto al existir cientos de aplicaciones por desarrollar, se prevé un aumento en el uso y aceptación del marketing virtual.

El Marketing virtual estará limitado solamente por la imaginación de quienes diseñan las aplicaciones. Al igual que los libros, la radio, la televisión y las computadoras, un medio puede servir como base para un interminable flujo de ideas y productos.

6.2.2 Realidad aumentada

El comercio electrónico y la publicidad online se verán afectados positivamente por la Realidad Aumentada (AR). La tecnología cambia la forma de ver el mundo, pero la Realidad Aumentada va un poco más allá, añade una capa sensorial tangible, añade volumen y movimiento a las imágenes. La Realidad Aumentada es la fusión del mundo real con el mundo virtual, en donde los objetos gráficos se mezclan en imágenes reales en tiempo real. La realidad aumentada crea la ilusión de que los objetos o información virtuales generados por el computad o el teléfono móvil son reales. La Realidad Aumentada tendrá una aplicación práctica en sectores estratégicos en donde la ubicación física de las cosas y su contexto son muy importantes:

- Medicina: sistemas de monitorización.
- Seguridad y defensa: entrenamiento, recreación de escenarios, simulación
- Arquitectura: proyección, diseño, maquetación
- Negocio inmobiliario: demostración
- Turismo: demostración de hoteles y destinos, recreaciones en museos y monumentos históricos.
- Entretenimiento: juegos.

Esta tecnología es útil en comercio electrónico principalmente en la demostración de productos y servicios. Se tiene gran expectativa en sectores B2C como turismo online (destinos, hoteles, restaurantes), comercio minorista en general (en catálogos virtuales de productos) y también en sectores industriales en donde sea necesaria la demostración de procesos y productos como en comercio exterior para visualizar productos de exportación - importación.

6.3 Las redes sociales y Facebook

Las redes sociales también están sirviendo de plataforma para el Comercio Electrónico, debido a la numerosa cantidad de usuarios que es posible alcanzar por

medio de éstas. Se empieza a hablar de un nuevo término “Social Commerce” o Comercio Electrónico Social. Una de las redes sociales más destacadas en Comercio Electrónico es Facebook www.facebook.com. Permite la venta de productos y servicios mediante catálogos y posee un servicio de pago en línea propio de Facebook. El f-Commerce (facebook Commerce)[18] tiene un gran potencial debido a la capacidad de segmentación de sus usuarios, el gran número de usuarios y el alto crecimiento de esta red social, etc.

6.4 Dinero digital

El dinero digital consiste en una nueva tecnología que aparece mostrando un gran potencial a futuro, puesto que podría cambiar la forma en la que se realizan los pagos al menudeo y por pequeños montos, por lo que pudiera conseguirse un impacto tan grande como el que trajo la introducción de la tarjeta de crédito.

La transición entre el papel moneda y la moneda electrónica pudiera darse a través del uso de tarjetas inteligentes, las mismas que guardan el saldo del dueño en sus componentes electrónicos.

6.4.1 Tarjetas inteligentes

La tarjeta inteligente ha sido definida de muchas formas, pero generalmente se la define como “un dispositivo de almacenamiento de datos con inteligencia (memoria en chip) y que aprovisiona identidad y seguridad”. Estas tarjetas son pequeñas microcomputadoras que no tienen una fuente de poder externa, pantalla ni teclado.

El chip de microprocesador en una tarjeta inteligente es especializado y tiene un diseño personalizado, generalmente con un control de patente específico. Ciertos datos, principalmente relacionados a la seguridad de la tarjeta, pueden ser ingresados solamente en la fabricación de la tarjeta. Adicionalmente al microprocesador cada tarjeta generalmente tiene varios kilobytes de memoria permanente, tanto de tipo re-escribible y no re-escribible.

La seguridad de las tarjetas inteligentes difiere de las tarjetas de banda magnética. La seguridad incluye características tanto de software como de hardware. La protección de software se basa en el control del acceso a los datos y el uso de técnicas de encriptación y las protecciones de hardware son realizadas durante la elaboración de la tarjeta inteligente. Las tarjetas inteligentes usan encriptación para llevar a cabo muchas tareas, incluyendo autorizaciones de pagos.

Es importante recordar que la decisión final al realizar cualquier transacción es hecha por el microprocesador de la tarjeta inteligente, lo cual es completamente

[18] (Comercio Electronico Global 2010)

diferente del proceso que realiza la tarjeta de crédito, por tanto una transacción en línea o fuera de línea es básicamente la misma para una tarjeta inteligente. La seguridad de una tarjeta inteligente debe ser más sofisticada debido a que la tarjeta nunca tiene que revelar la contraseña de su dueño a un sistema externo.

La seguridad mejorada de las tarjetas basadas en chip es notable debido a que la aplicación principal de las tarjetas de banda magnética son las tarjetas financieras o bancarias. Estas tarjetas han conducido a muchos nuevos tipos de crímenes.

La vida útil de una tarjeta inteligente es aproximadamente cinco años, en contraste con la tarjeta magnética cuya vida útil es de sólo dos a tres años. En cambio el costo de fabricación de la tarjeta inteligente es mayor que el de la tarjeta de cinta magnética.

Tanto la tarjeta de cinta magnética y la tarjeta inteligente han servido como tarjetas de prepago. El uso más común para tarjetas prepago son las transacciones específicas que requieren el pago de un monto pequeño y fijo. Ejemplos de uso de tarjetas de prepago son: llamadas telefónicas, peaje en carreteras, boletos para transporte público, etc., las mismas que son de bajo valor monetario. Por su parte, para los consumidores, este tipo de tarjetas puede resultar más cómodo y seguro que llevar monedas o billetes. Las tarjetas de prepago para propósitos específicos no son sujetas a ninguna regulación financiera. Pueden compararse como la compra de fichas por adelantado para la adquisición de un bien o de un servicio. Por su parte, las tarjetas inteligentes recargables tienen un funcionamiento más complicado.

Una ventaja considerable de las tarjetas inteligentes sobre las tarjetas de banda magnética tiene que ver con su creciente funcionalidad. Las capacidades de almacenamiento pudieran permitir el uso de tarjetas inteligentes para uso en diversos campos como en medicina, las mismas que contendrían la información médica del dueño, la misma que es vital ya sea en caso de revisiones de rutina o de emergencias.

A pesar de que la capacidad de memoria de las tarjetas inteligentes es limitada, se está planificando mayores niveles de almacenamiento para las próximas generaciones. Al tener mayor capacidad de memoria, las aplicaciones para las tarjetas inteligentes serían mucho mayores.

Existe además, otro tipo de tarjeta, la tarjeta óptica o tarjeta láser, la misma que ha sido utilizada en determinadas aplicaciones en países como Japón. La tarjeta óptica tiene una memoria WORM (Write Once Read Many) no borrable. La capacidad de memoria de almacenamiento disponible es mucho mayor, llega hasta 4 megabytes,

y su uso tiene varias aplicaciones prácticas especialmente en el campo de la medicina. La tabla 6.1 resume las características de los tipos tarjetas electrónicas más utilizadas.

Tipo de tarjeta	**Capacidad**	**Seguridad**
Tarjeta telefónica	10 bytes	Ninguna
Plástica	Cientos de bytes	Ninguna
Cinta magnética	Cientos de bytes	Encriptación limitada. La protección de los datos la realizan dispositivos externos
Tarjeta inteligentes	Varios kilobytes	Protección brindada por su propio microprocesador
Tarjeta ópticas	Varios megabytes	Encriptación limitada. La protección de los datos la realizan dispositivos externos

Tabla 6.1 Tipos de tarjetas electrónicas

DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

7.1 Análisis de requerimientos

7.1.1 Diagrama de Casos de Uso de la Tienda Virtual

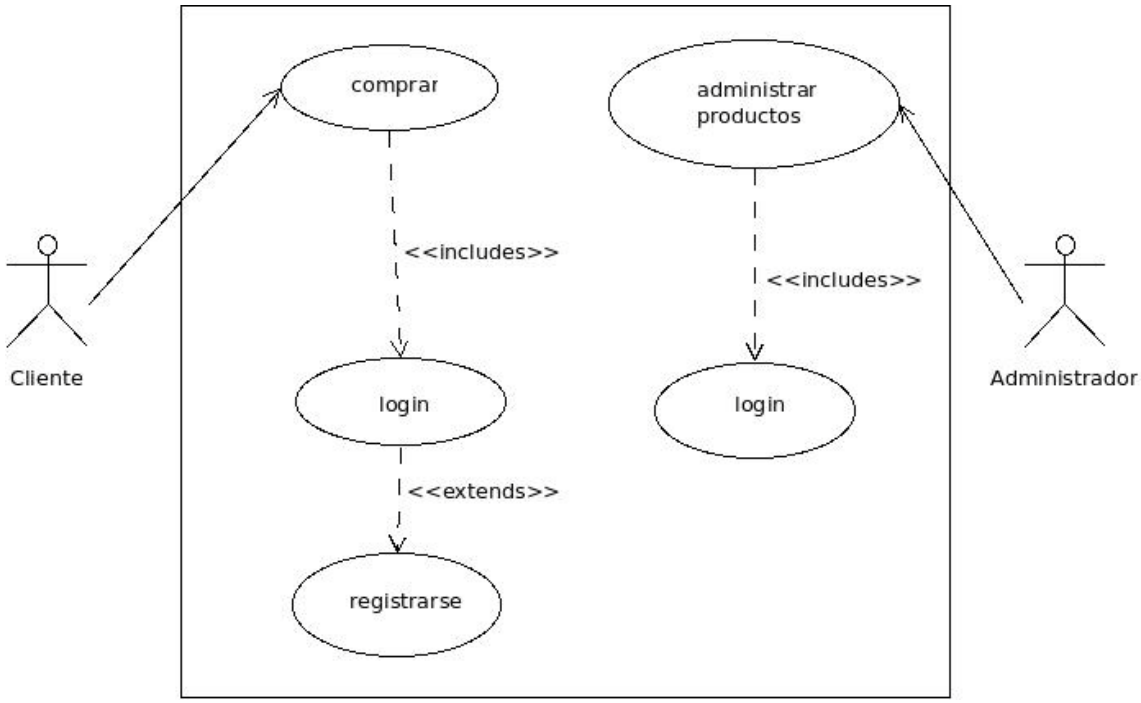

Figura 7.1 Diagrama de casos de uso de la tienda virtual

7.1.2 Especificaciones de los Casos de Uso

7.1.2.1. Caso de Uso: Registrarse

Resumen: el visitante ingresa sus datos en el sistema.
CURSO NORMAL DE EVENTOS:

1. El visitante escoge la opción registrarse
2. El sistema muestra el formulario de registro
3. El visitante ingresa su información: nombres, apellidos, cédula (si es ecuatoriano), dirección, teléfono, celular (opcional), email principal, email secundario (opcional), ciudad, provincia, país, fecha de nacimiento (opcional), contraseña.
4. El sistema guarda la información del visitante y envía un email al correo del visitante con el código de activación de la cuenta.
5. El visitante revisa su correo elctrónico e ingresa el código de activación de la cuenta.
6. El sistema verifica el código de activación y activa la cuenta del visitante

CURSO ALTERNATIVO DE EVENTOS

3.1 La dirección de correo electrónico ingresada por el visitante ya consta en la base de datos.
3.1.1 El sistema muestra un mensaje indicando que el correo electrónico ya está registrado y muestra la pantalla para ingresar al sistema
3.2 El visitante no ingresa todos los datos obligatorios
3.2.1 El sistema muestra un mensaje indicando que se deben llenar los datos obligatorios.
3.3 El visitante ingresa una dirección de correo electrónico no válida
3.3.1 El sistema muestra un mensaje indicando que la dirección de correo electrónico no es válida
3.4 Si la contraseña ingresada tiene menos de 5 caracteres, el sistema muestra un mensaje de que la contraseña es muy corta y que el visitante tiene que ingresar una contraseña más grande.
6.1 El código de activación de la cuenta es incorrecto.
6.1.1 El sistema indica al visitante que el código ingresado es incorrecto

Precondiciones:

- El visitante cuenta con una dirección de correo electrónico de cualquier servidor en Internet
- El visitante ha escogido el idioma

Postcondiciones:

El visitante se ha registrado en la tienda virtual

Actores: Visitante de la tienda virtual.

7.1.2.2 Caso de Uso: Login

Resumen: el visitante ingresa a su cuenta en la tienda virtual

CURSO NORMAL DE EVENTOS

1. El visitante escoge ingresar a su cuenta en la tienda virtual
2. El sistema muestra una pantalla pidiendo la dirección de email y la contraseña
3. El visitante ingresa la dirección de email y la contraseña y presiona el botón enviar.
4. El sistema verifica los datos y concede acceso al visitante dependiendo de si es un cliente de la tienda virtual o administrador de la tienda virtual y muestra la pantalla correspondiente

CURSO ALTERNATIVO DE EVENTOS

4.1 El sistema encuentra que la dirección de email y la contraseña ingresadas no concuerdan con la información almacenada en la base de datos y muestra el mensaje de error.
4.2 El usuario ha olvidado la contraseña y accede a la opción: “Contraseña olvidada”
4.2.1 El sistema muestra la pantalla pidiendo el correo electrónico del visitante.
4.2.2 El visitante ingresa la dirección de correo electrónico
4.2.3 El sistema busca la dirección de correo electrónico del visitante y si la encuentra envía un email con un enlace para ingresar la nueva contraseña.
4.2.3.1 En caso de que el sistema no encuentre la dirección de correo electrónico
ingresada, entonces muestra una pantalla indicando que la dirección de correo
electrónico no ha sido registrada en el sistema
4.2.4 El visitante ingresa a su cuenta de correo electrónico revisa el correo enviado por el sistema y accede al enlace para establecer la nueva contraseña
4.2.5 El visitante ingresa la nueva contraseña
4.2.6 El sistema registra la nueva contraseña asociada a la dirección de correo
electrónico del visitante.

PRECONDICIONES

1. El visitante ha registrado una cuenta en la tienda virtual
2. El visitante ha escogido el idioma

POSTCONDICIONES

1. El visitante ha accedido a su cuenta en la tienda virtual

ACTORES: visitante, administrador de la tienda virtual, sistema

7.1.2.3 Caso de uso: comprar

Resumen: el visitante realiza la compra de productos en la tienda virtual

CURSO NORMAL DE EVENTOS

1. El visitante revisa los productos de la tienda virtual
2. El visitante presiona el botón “comprar” asociado a un producto
3. El sistema muestra el carro de compras en la parte superior de la pantalla el mismo que detalla el producto, la cantidad de unidades (al inicio es 1), el precio y el total y muestra el botón “Registrar compra”
4. El visitante sigue explorando la tienda virtual agrega más productos al carro de compras.
5. El sistema actualiza el carro de compras y detalla los productos agregados y calcula el total de la compra.
6. El visitante decide borrar un producto del carro de compras, para lo cual presiona el botón quitar del carro de compras
7. El sistema actualiza el carro de compras sin los datos del producto eliminado y calcula el total.
8. El visitante actualiza la cantidad a comprar de un producto en el carro de compras
9. El sistema calcula el total con los datos actualizados
10. El visitante presiona el botón registrar compra
11. El sistema muestra la pantalla con la opción para ingresar al sistema
12. El usuario ingresa su dirección de correo electrónico y su contraseña
13. El sistema muestra el carro de compras y los datos del cliente en los que se incluye la dirección , ciudad, país, el teléfono del cliente; y calcula los cargos por concepto de transporte dependiendo de la dirección del visitante.
14. El visitante escoge la forma de pago
15. En caso de que la forma de pago escogida sea con tarjeta de crédito, entonces el sistema muestra la pantalla perteneciente a la pasarela de pago
16. El visitante ingresa los datos de la tarjeta de crédito
17. La pasarela de pagos realiza el cargo a la tarjeta de crédito y asigna la cantidad a la cuenta de comerciante de la tienda virtual.
18. El sistema registra la transacción como pagada y registra el estado de los productos como no entregados.
19. El sistema notifica al administrador del sistema por medio de correo electrónico que una nueva venta se ha efectuado.
20. El administrador del sistema verifica la venta, revisa que haya sido cancelada, realiza la verificación de la validez de los datos y envía el producto al destinatario a su dirección de domicilio.

CURSO ALTERNATIVO DE EVENTOS

8.1 Si la cantidad ingresada por el visitante supera a la cantidad en existencia del producto, el sistema muestra un mensaje la cantidad existente y realiza el cálculo del total de acuerdo a esta cantidad.

12.1 En caso de que el visitante no esté registrado en la tienda virtual, el sistema muestra la pantalla de registro de la tienda virtual

13.1 En caso de que los datos del visitante no estén actualizados, el visitante actualiza los datos como dirección, ciudad, país, teléfono.

15.1 En caso de que el método de pago sea diferente al de tarjeta de crédito, entonces el sistema registra la transacción como “no pagada” y registra el estado de los productos como “no enviados”

17.1 La pasarela de pagos encuentra que la tarjeta de crédito ingresada tiene problemas (no tiene crédito, ha sido reportada como robada o perdida, tiene valores por cancelar, etc.). El sistema indica que no pudo realizar la venta con la tarjeta de crédito ingresada y registra la transacción como “no pagada”.

PRECONDICIONES:

El visitante ha leído y está de acuerdo con las condiciones de uso de la tienda virtual.

POSTCONDICIONES:

El visitante ha realizado la compra de 1 o varios productos en la tienda virtual.

Actores: visitante, administrador de la tienda virtual, carro de compras, pasarela de pagos.

Otra información:

Formas de pago admitidas:

- Tarjeta de crédito
- Transferencia por Western Union
- Pago en efectivo al momento de la entrega del producto (válido solamente en la ciudad de Loja)

Pasarela de pagos: Payment gateway, software asociado a los bancos que verifica la validez de las tarjetas de crédito y se encarga de realizar el cobro a las mismas.

7.1.2.4 Caso de uso: Administrar productos

Resumen: el administrador/operador de la tienda virtual ingresa o actualiza los productos que están a la venta

CURSO NORMAL DE EVENTOS

1. El administrador del sistema ingresa a su cuenta en la tienda virtual, ingresa dirección de email y contraseña
2. El sistema verifica los datos ingresados y concede el acceso al administrador.
3. El administrador escoge la opción administrar productos
4. El sistema muestra el listado de productos registrados
5. El administrador escoge un producto para actualizar o escoge la opción de ingresar un producto nuevo.
6. El sistema muestra la pantalla de edición de productos.
7. El administrador ingresa el título del producto (español e inglés), la descripción del producto (español e inglés), el precio, la cantidad existente, el peso, fotografías del producto, las categorías a las que pertenece.
8. El sistema guarda los datos ingresados.

CURSO ALTERNATIVO DE EVENTOS

8.1 El sistema encuentra que el producto ingresado ya consta en el sistema. Por tanto muestra el mensaje correspondiente.

PRECONDICIONES

El administrador/operador del sistema ha sido registrado en la tienda virtual

POSTCONDICIONES

El administrador/operador ha ingresado o actualizado los datos de un producto

Actores: administrador/operador de la tienda virtual

7.2 Diseño Lógico de la Tienda Virtual

7.2.1 Diagramas de Secuencia

7.2.1.1 Diagrama de Secuencia: Registrarse

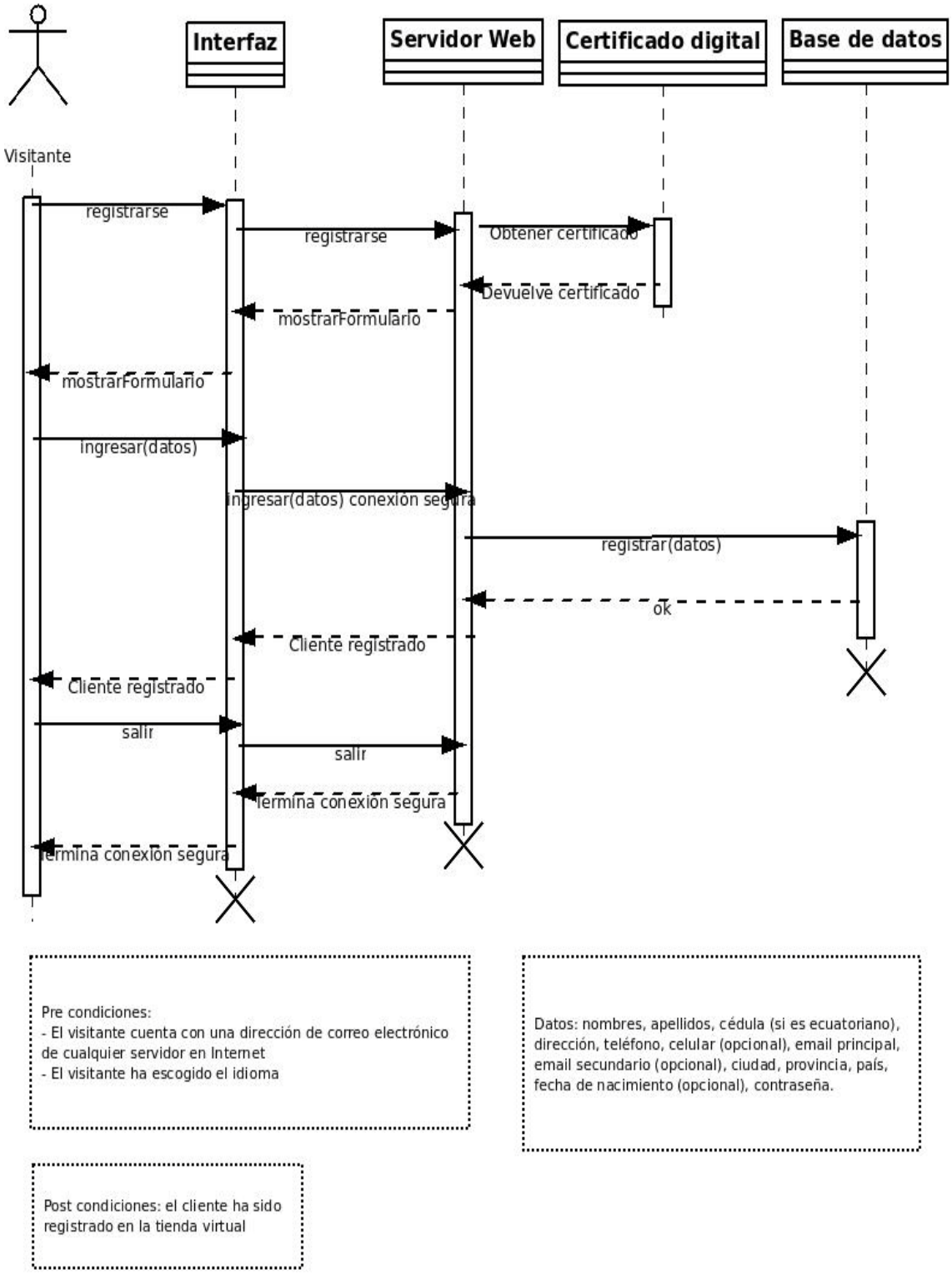

Figura 7.2 Diagrama de secuencia - Registrarse

7.2.1.2 Diagrama de Secuencia: Login

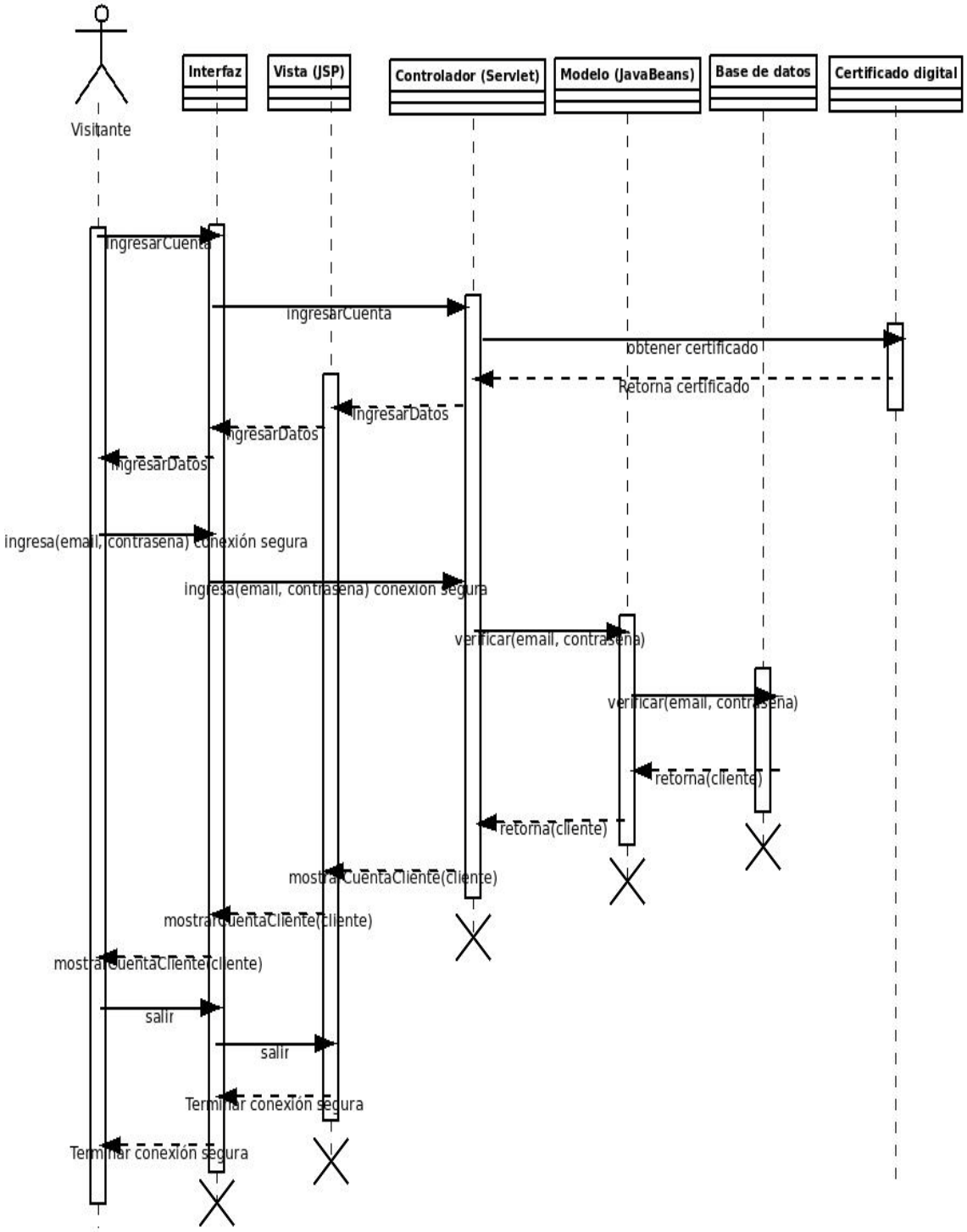

Figura 7.3 Diagrama de secuencia - Login

7.2.1.3 Diagrama de Secuencia: Comprar

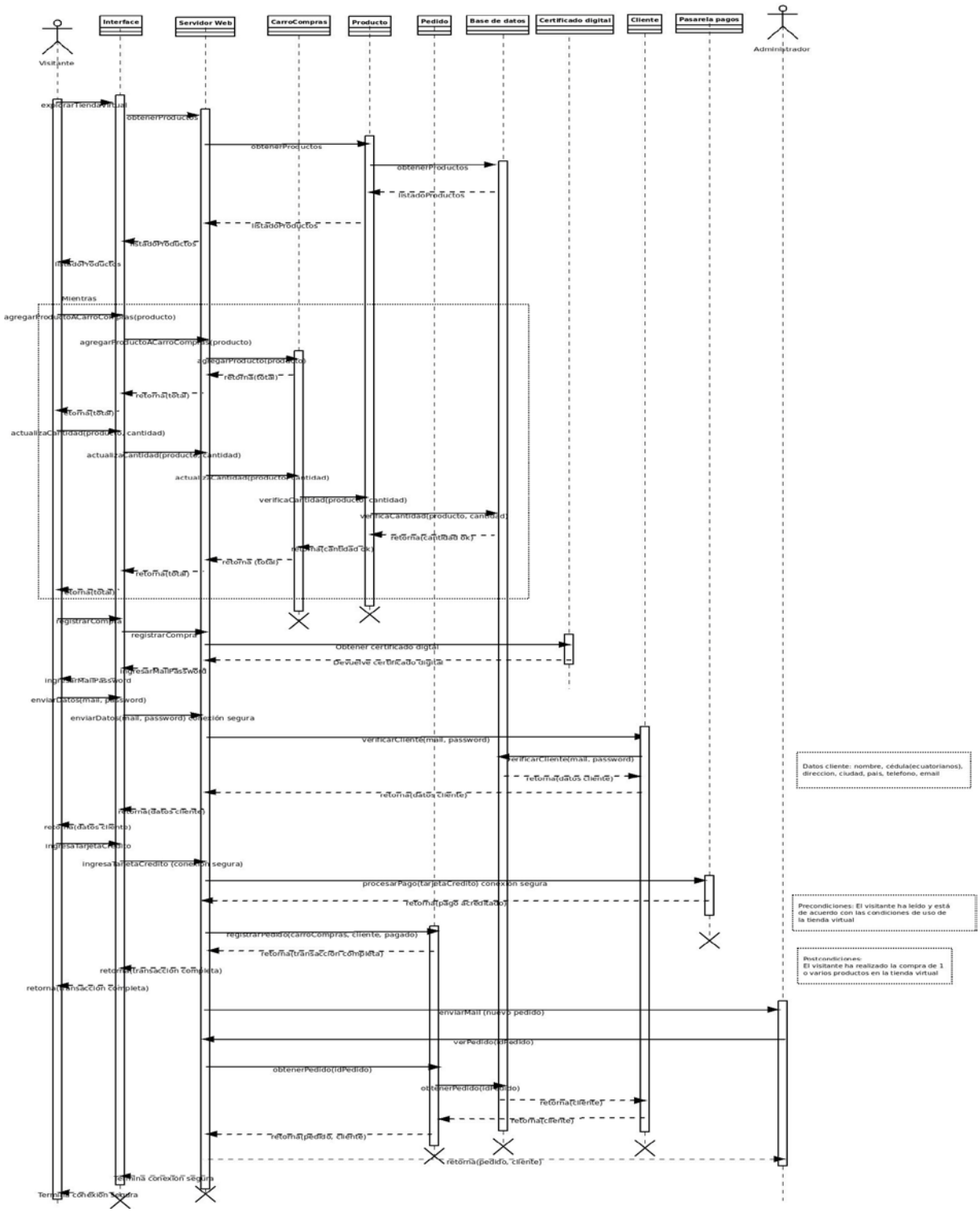

Figura 7.4 Diagrama de secuencia - Comprar

7.2.1.4 Diagrama de Secuencia: Administrar productos

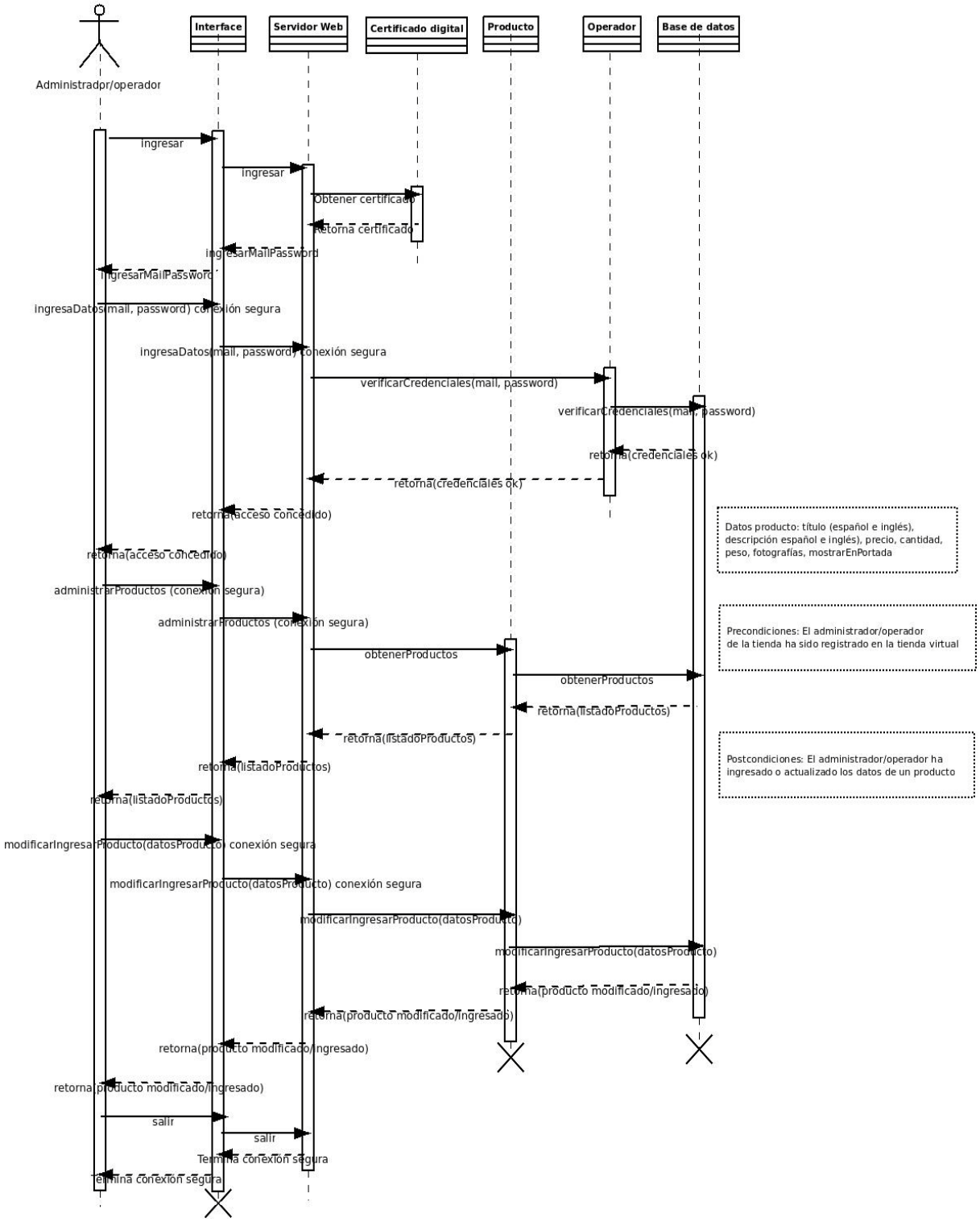

Figura 7.5 Diagrama de secuencia – Administrar productos

7.2.2 Diagramas de Colaboración

7.2.2.1 Diagrama de colaboración: Registrar cliente

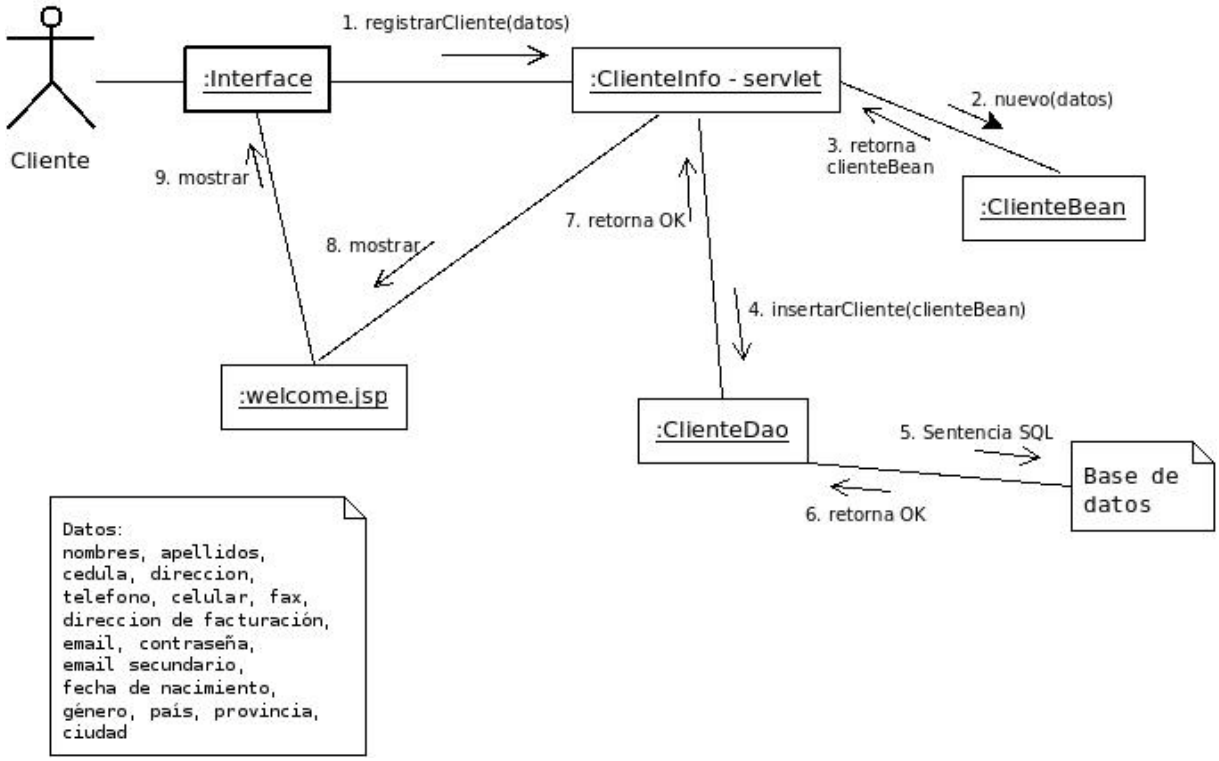

Figura 7.6 Diagrama de colaboración – Registrar cliente

7.2.2.2 Diagrama de colaboración: Login

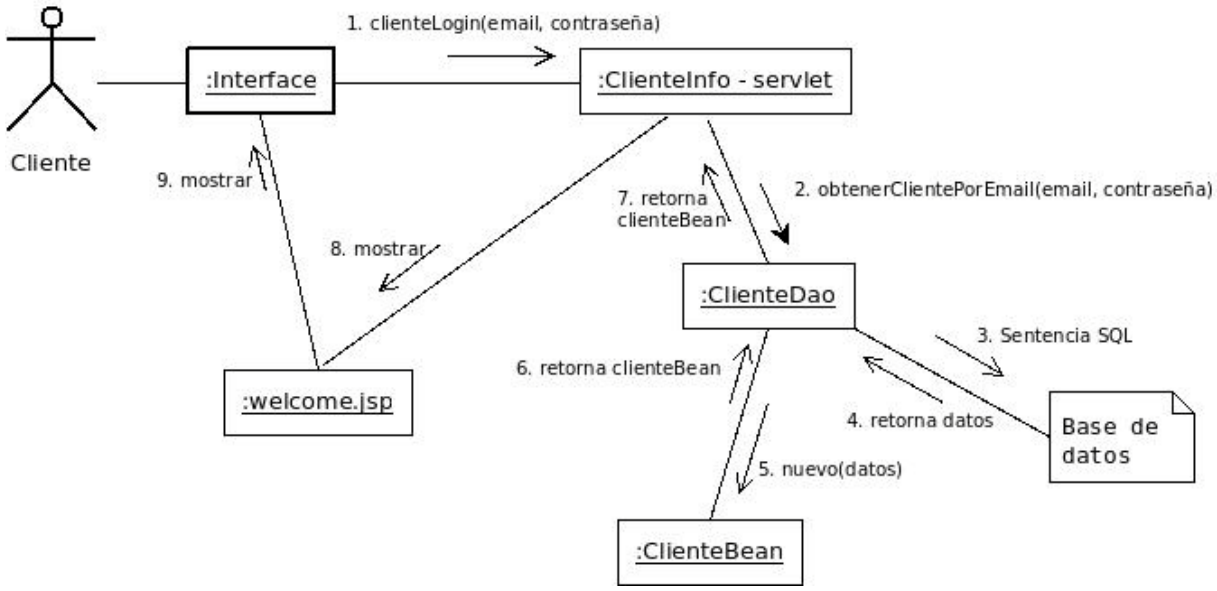

Figura 7.7 Diagrama de colaboración – Login

7.2.2.3 Diagrama de colaboración: Administrar productos

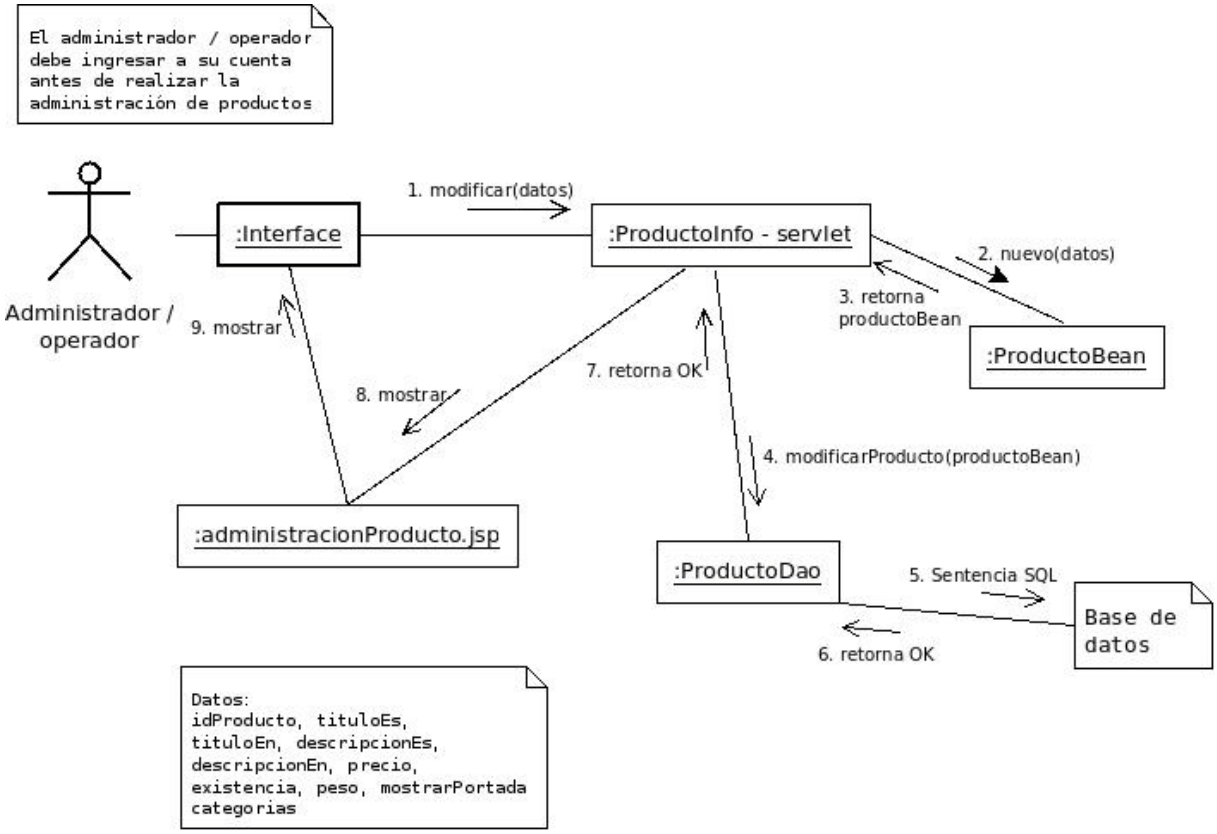

Figura 7.8 Diagrama de colaboración – Administrar productos

7.2.2.4 Diagrama de colaboración: Comprar

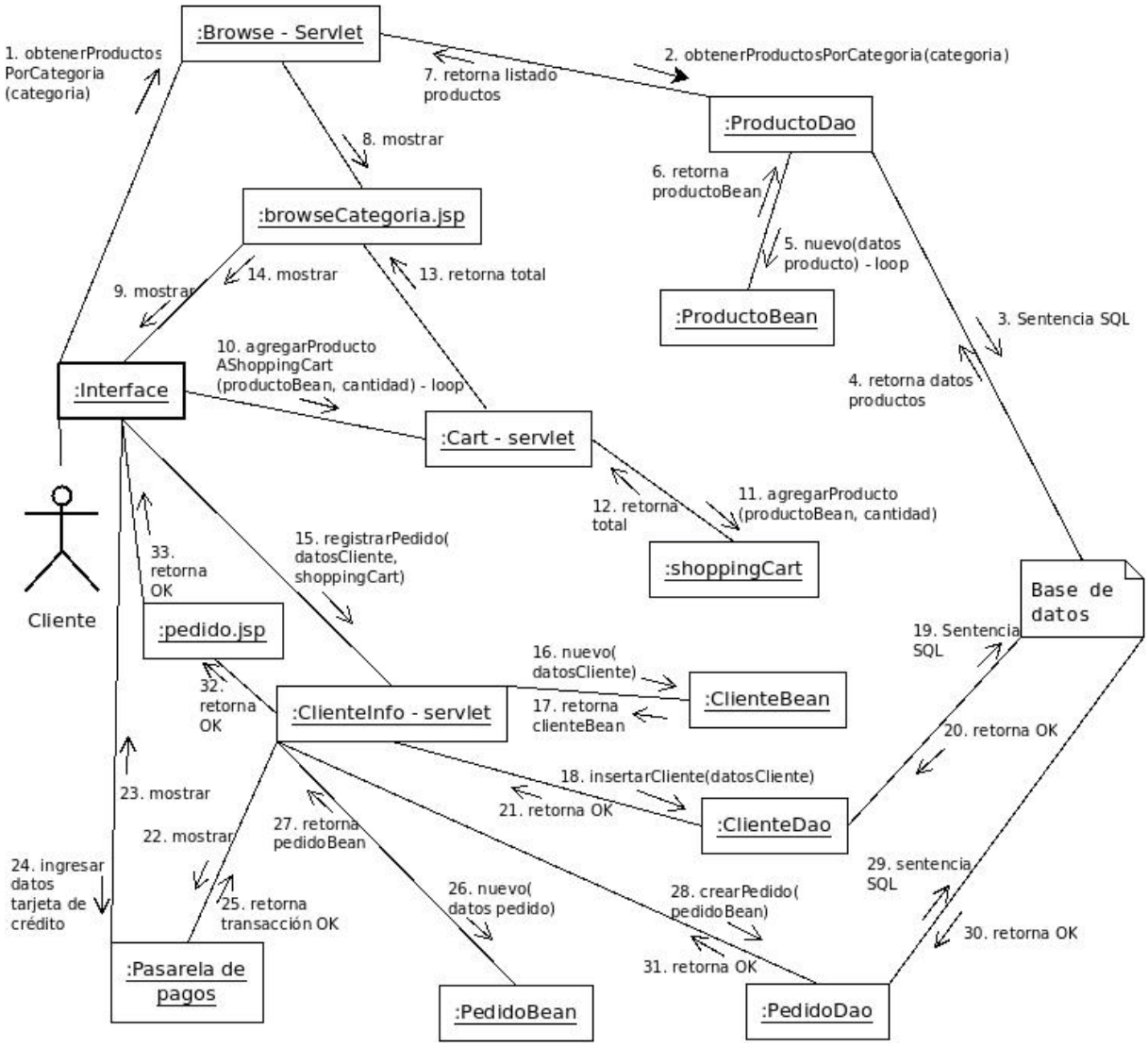

Figura 7.9 Diagrama de colaboración – Comprar

7.2.3 Diagrama de Clases

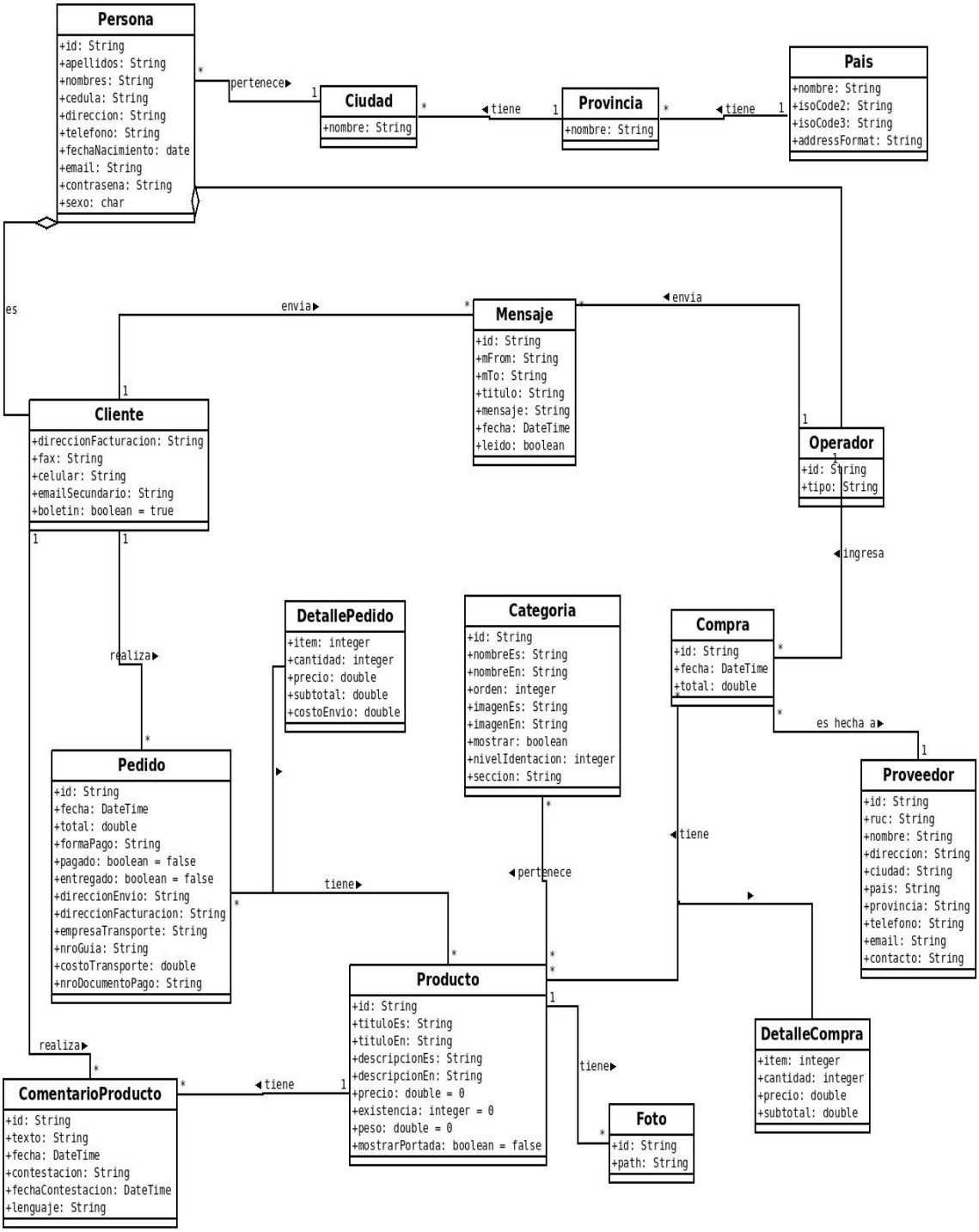

Figura 7.10 Diagrama de clases

7.2.4 Diagrama Entidad – Relación

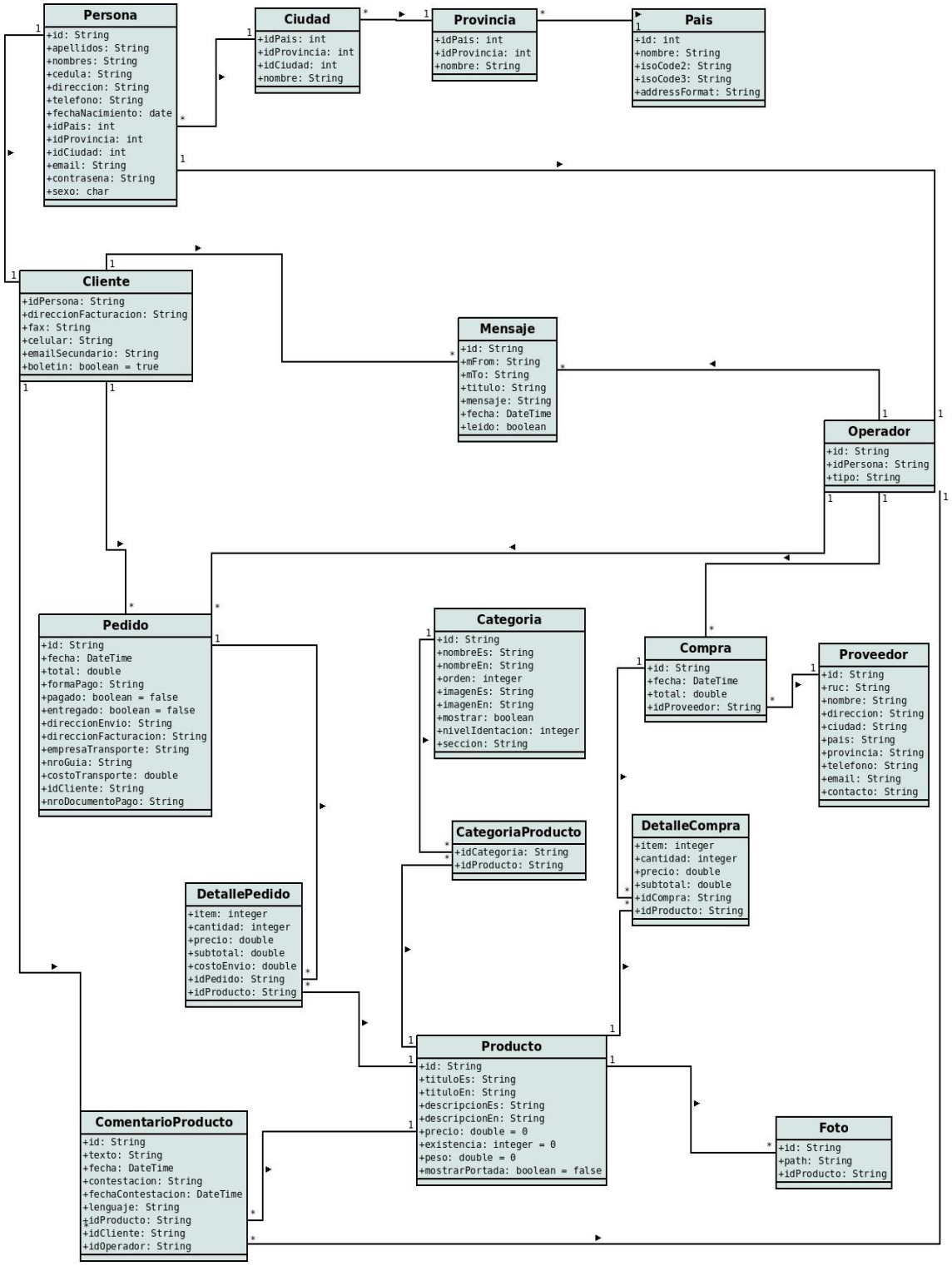

Figura 7.11 Diagrama Entidad - Relación

7.2.5 Diagrama de Estados

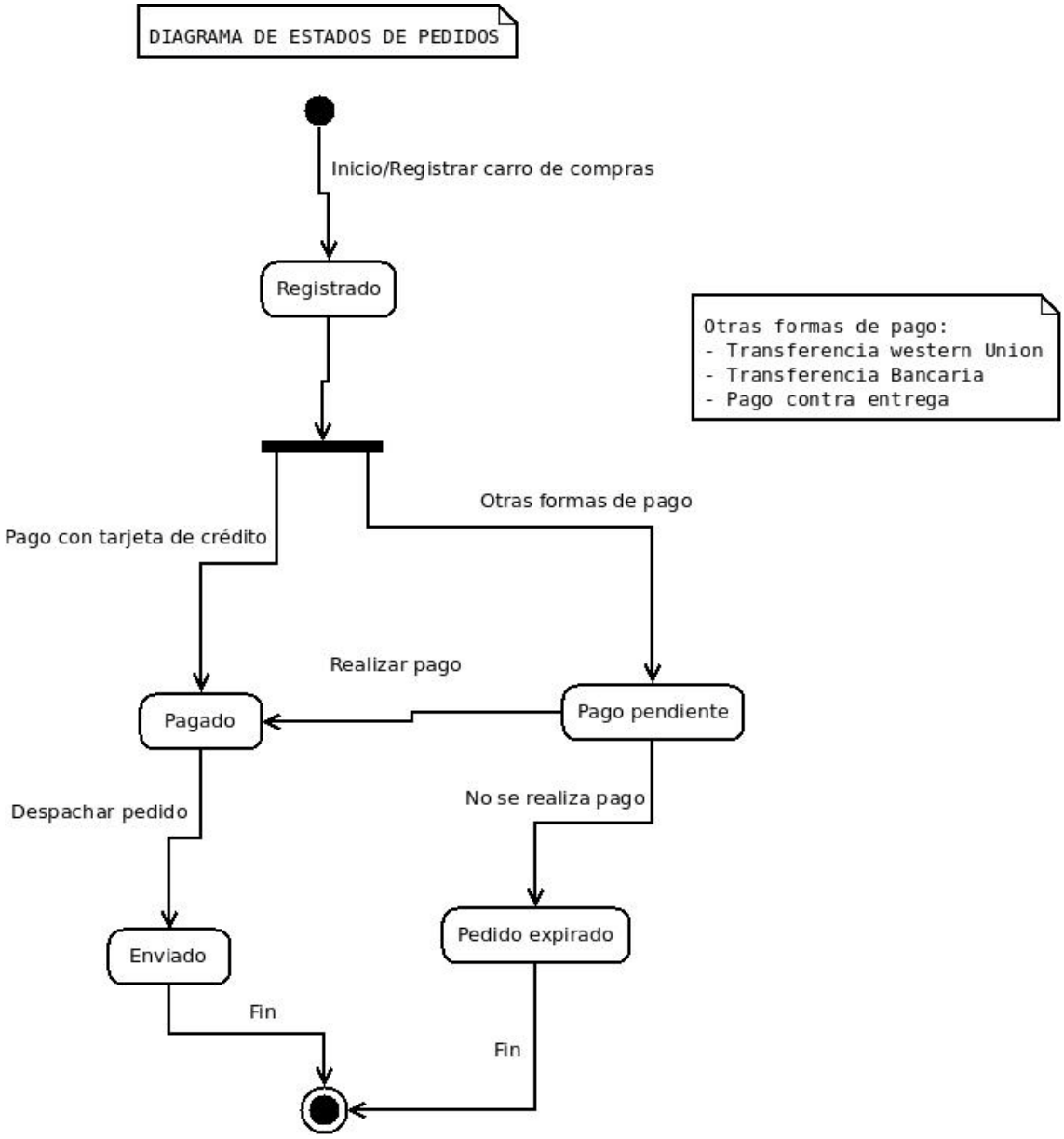

Figura 7.12 Diagrama de estados

7.2.6 Diagrama de Componentes

7.2.6.1 Diagrama de Componentes General

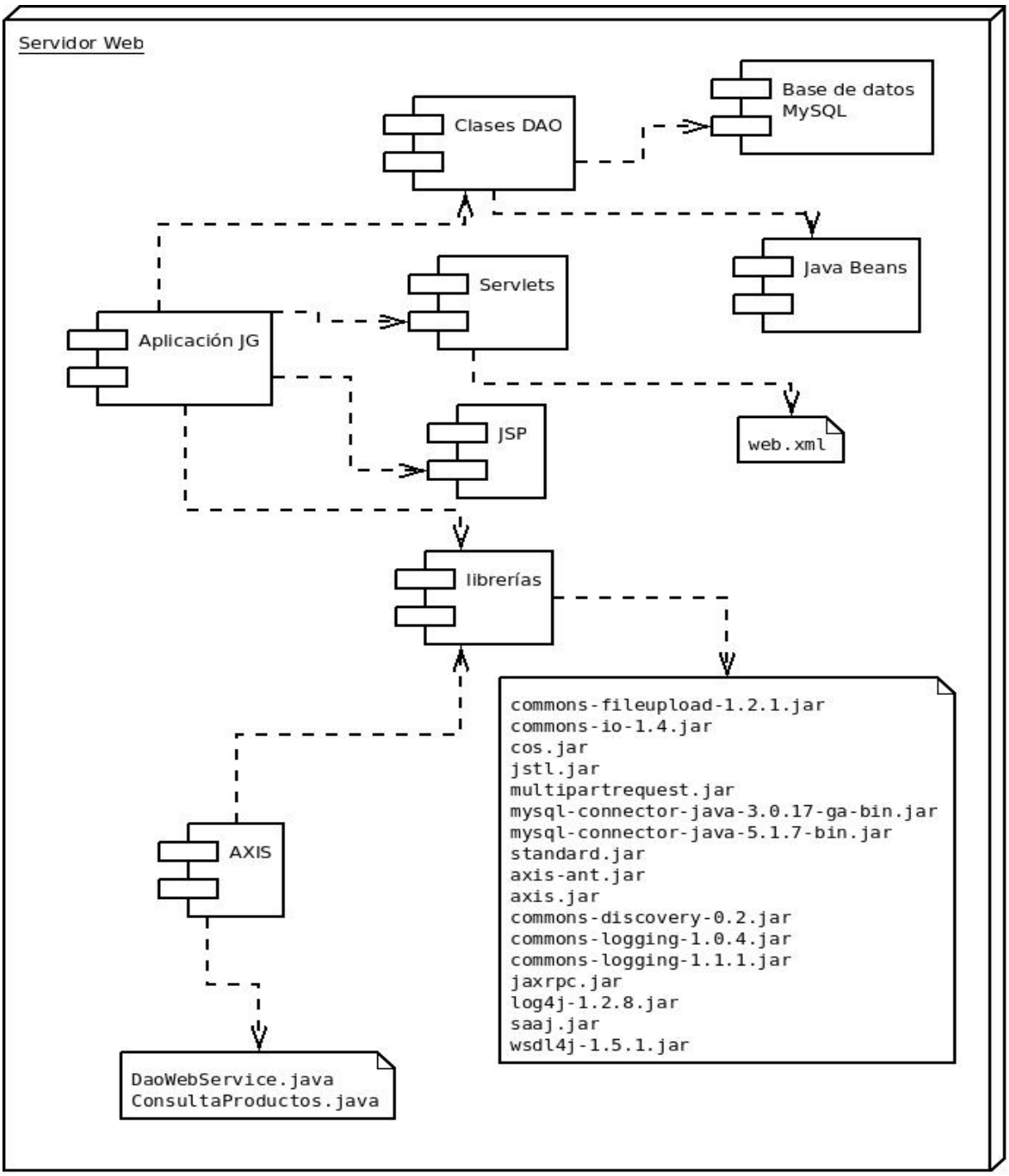

Figura 7.13 Diagrama de componentes general

7.2.6.2 Diagrama de Componentes detallado

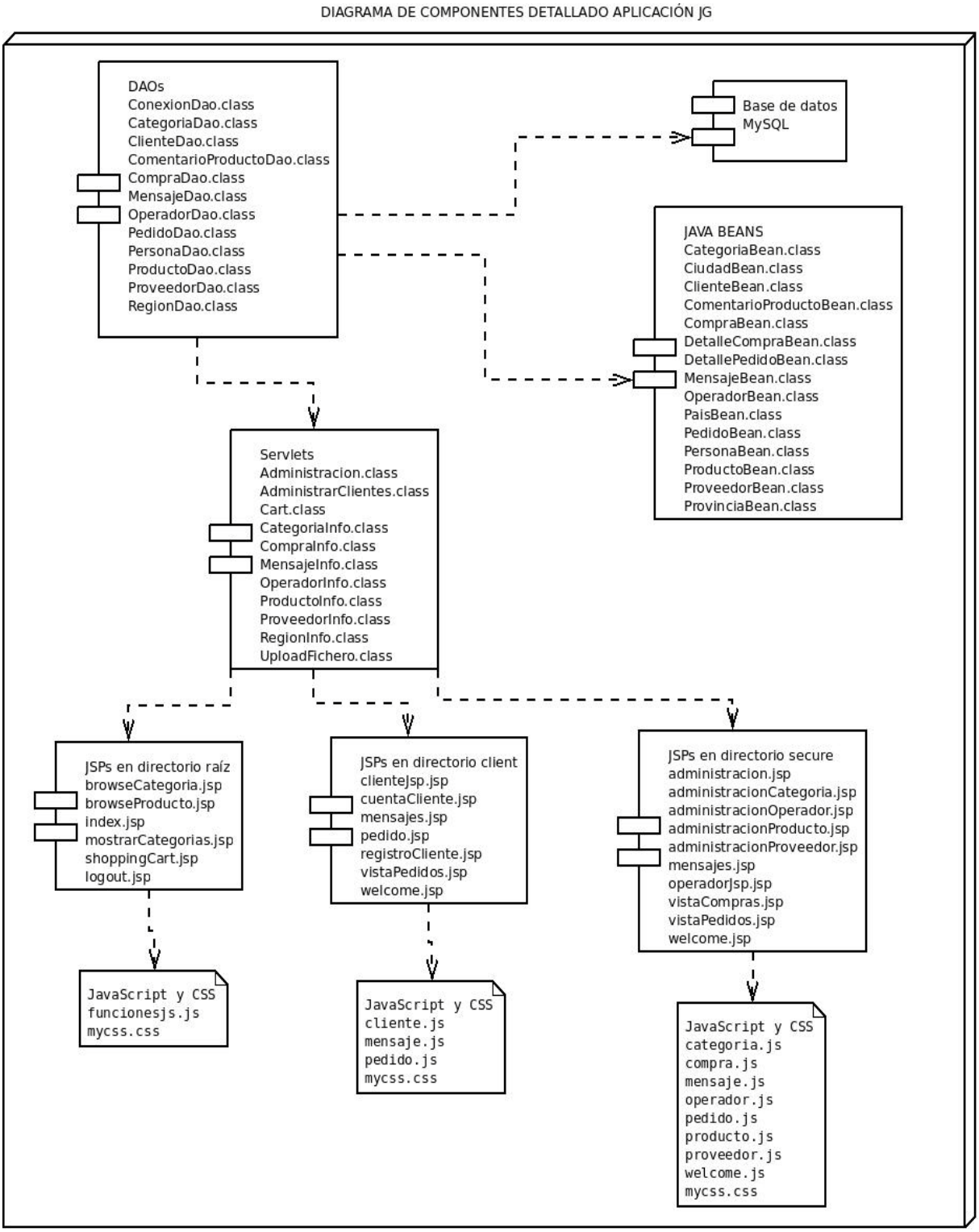

Figura 7.14 Diagrama de componentes detallado

7.2.7 Diagramas de Distribución

7.2.7.1 Diagrama de Distribución

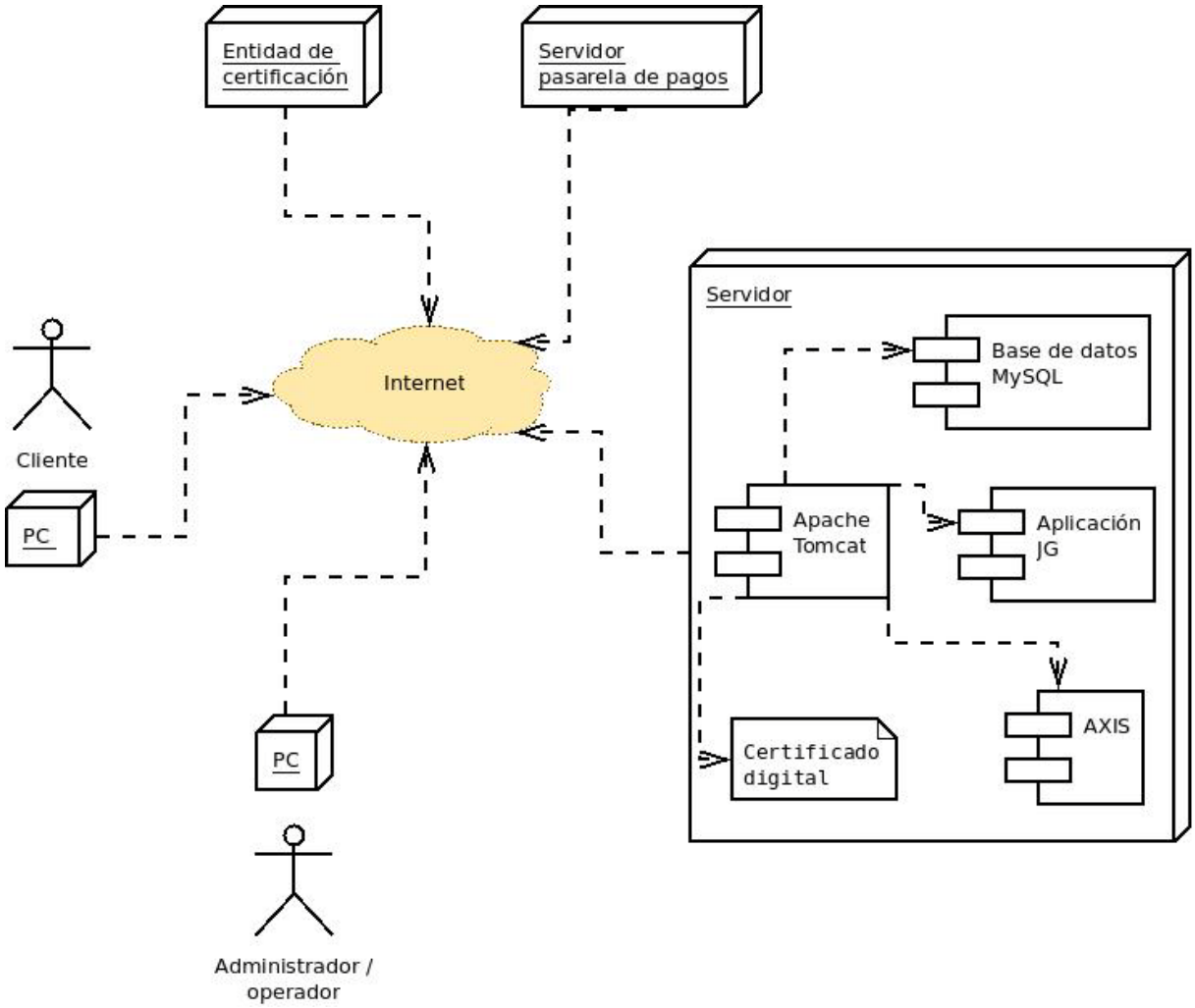

Figura 7.15 Diagrama de distribución

7.2.7.2 Diagrama de Distribución de Servicios Web

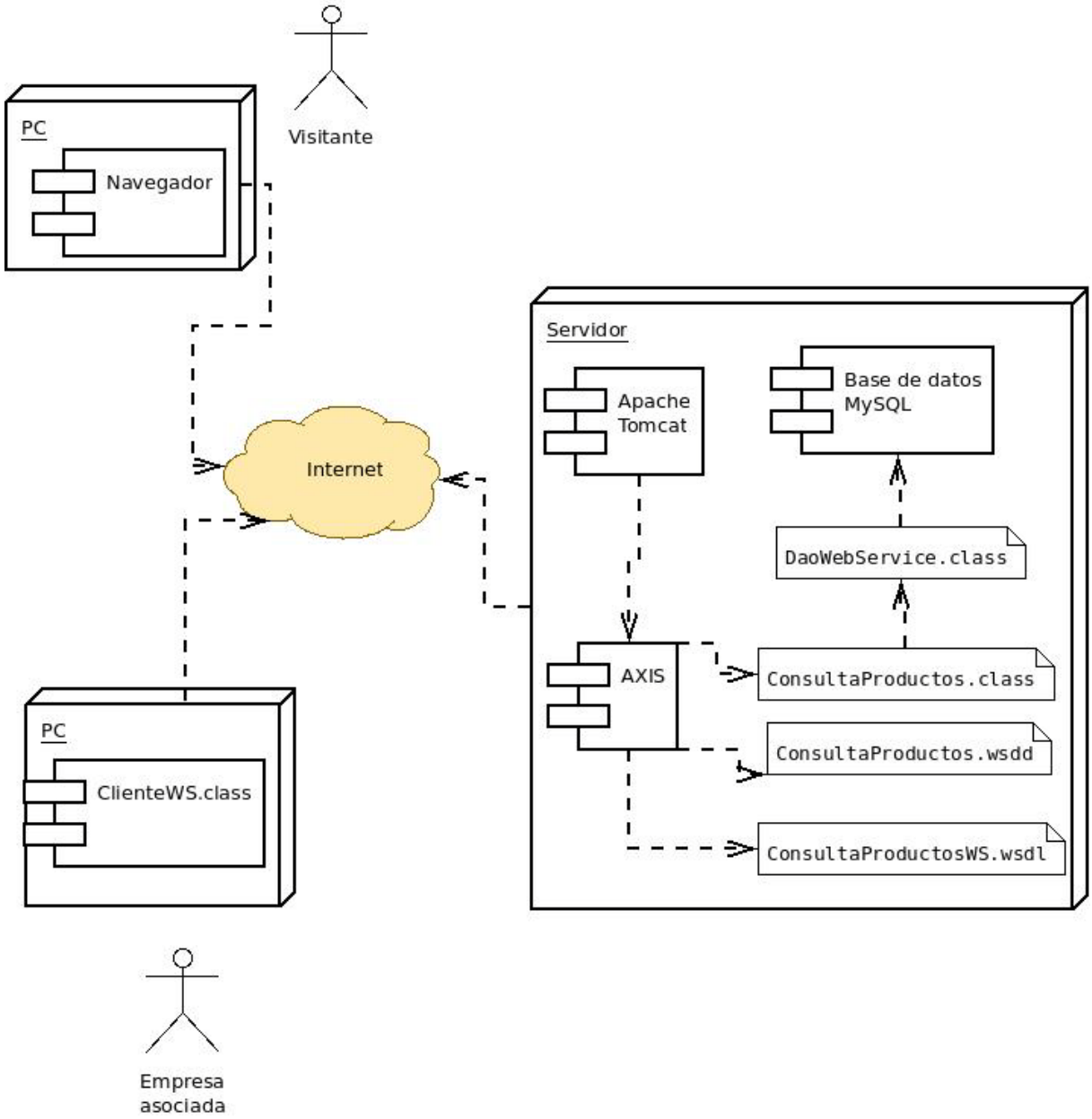

Figura 7.16 Diagrama de distribución de Servicios Web

7.3 Codificación

Para la codificación de la tienda virtual se utilizó el entorno de desarrollo NetBeans versión 6.8. En lo referente al servidor se utilizó el contenedor Web Tomcat 6.0 hospedado en el sistema operativo Ubuntu 10.04.1. Para la implementación de servicios web se utilizó la plataforma Axis 1.4.

Por otra parte, en cuanto al cliente, se utilizó el navegador web Firefox 3.0.19.

Se utilizó el modelo por capas MVC (Modelo – Vista – Controlador), en el que el Modelo representado por la tecnología JavaBeans representa las estructuras que componen el negocio, el Controlador representado por la tecnología de Servlets que contiene la lógica del negocio y finalmente la Vista representada por la tecnología JSP que tiene que ver con la interfaz con el usuario. Cabe mencionar que las clases JavaBeans trabajan en conjunto con clases agrupadas en el paquete “daos” (DAO = Data Access Object), estas clases son las que contienen las sentencias SQL que interactúan con la base de datos.

Para el diseño gráfico de la tienda virtual de Comercio Electrónico se requirió la ayuda de la tecnología CSS (Cascade Style Sheets), mediante la cual se logró desarrollar una plantilla que sirvió como base para todas las pantallas de la aplicación compuestas por las páginas JSP, mediante esta implementación se consigue ahorrar tiempo y esfuerzo.

Se utilizó AJAX (Asynchronous Javascript and XML) para facilitar la navegación del usuario en la tienda virtual. Por medio de AJAX, no se refresca constantemente la pantalla cuando se agrega productos al carro de compras por lo que el tiempo de espera de una petición se reduce considerablemente, así mismo en la parte de administración, el operador se ve beneficiado de igual forma cuando realiza la gestión de productos, categorías, pedidos, compras, etc. Así también, cuando el operador registra una compra a un proveedor, AJAX permite a la aplicación dar sugerencias de productos mientras el operador digita los nombres de los productos, todo esto gracias a llamadas asincrónicas hacia la base de datos por parte de AJAX.

En cuanto a la implementación de servicios Web, se dio principal énfasis en la búsqueda de productos, siendo así se ha publicado un servicio Web con métodos para búsqueda avanzada de productos:

- Búsqueda por nombre del producto en español e inglés,
- Búsqueda de productos por categoría en español e inglés,

- Búsqueda de productos por rango de precios.

Estos métodos devuelven el listado de productos encontrados indicando existencias y precios.

El descriptor del servicio web publicado se lo puede encontrar en la dirección:

http://www.jgexclusividades.com:8080/axis/services/ConsultaProductosWS?wsdl

7.3.1 Cliente para consumo del servicio Web de la tienda virtual

ClienteService.java

```
import org.apache.axis.client.Call;

import org.apache.axis.client.Service;

import org.apache.axis.encoding.XMLType;

import javax.xml.rpc.ParameterMode;


public class ClienteService {

  public static void main(String [] args) throws Exception {

      String  endpoint =

"http://www.jgexclusividades.com:8080/axis/services/ConsultaProductosWS";

      String producto       = "carro";

      Service  service = new Service();

      Call     call    = (Call) service.createCall();

       // Establecemos la dirección en la que está activado el WebService

      call.setTargetEndpointAddress( new java.net.URL(endpoint) );

       // Establecemos el nombre del método a invocar

      call.setOperationName( "BuscarProductosIngles" );

       // Establecemos los parámetros que necesita el método
```

```
    // Observe que se deben especidicar correctamente tanto el nómbre como el tipo de datos..

    // esta información se puede obtener viendo el WSDL del servicio Web

    call.addParameter( "in0", XMLType.XSD_STRING, ParameterMode.IN );

    // Especificamos el tipo de datos que devuelve el método.

    call.setReturnType( XMLType.XSD_STRING );

    // Invocamos el método

    String result = (String) call.invoke( new Object [] { producto });

    // Imprimimos los resultados

    System.out.println("El resultado de la búsqueda es: " + result);

  }

}
```

7.3.2 Scripts SQL

7.3.2.1 Script de creación de tablas

```
DROP TABLE IF EXISTS TV_DETALLECOMPRA;
DROP TABLE IF EXISTS TV_COMPRA;
DROP TABLE IF EXISTS TV_DETALLEPEDIDO;
DROP TABLE IF EXISTS TV_PEDIDO;
DROP TABLE IF EXISTS TV_FOTO;
DROP TABLE IF EXISTS TV_CATEGORIAPRODUCTO;
DROP TABLE IF EXISTS TV_COMENTARIOPRODUCTO;
DROP TABLE IF EXISTS TV_COMENTARIO;
DROP TABLE IF EXISTS TV_MENSAJE;
DROP TABLE IF EXISTS TV_CATEGORIA;
DROP TABLE IF EXISTS TV_PROVEEDOR;
DROP TABLE IF EXISTS TV_PRODUCTO;
DROP TABLE IF EXISTS TV_OPERADOR;
DROP TABLE IF EXISTS TV_CLIENTE;
DROP TABLE IF EXISTS TV_PERSONA;
DROP TABLE IF EXISTS TV_CIUDAD;
DROP TABLE IF EXISTS TV_PROVINCIA;
DROP TABLE IF EXISTS TV_PAIS;


CREATE TABLE TV_PAIS(
```

```
  ID INT NOT NULL,
  NOMBRE VARCHAR(80) NOT NULL,
  ISOCODE2 VARCHAR(2) NOT NULL,
  ISOCODE3 VARCHAR(3) NOT NULL,
  ADDRESSFORMAT VARCHAR(5) NOT NULL,
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_PROVINCIA(
  IDPAIS INT NOT NULL,
  IDPROVINCIA INT NOT NULL,
  NOMBRE VARCHAR(80) NOT NULL,
  PRIMARY KEY(IDPAIS, IDPROVINCIA),
  FOREIGN KEY(IDPAIS) REFERENCES TV_PAIS(ID)
)TYPE=INNODB;

CREATE TABLE TV_CIUDAD(
  IDPAIS INT NOT NULL,
  IDPROVINCIA INT NOT NULL,
  IDCIUDAD INT NOT NULL,
  NOMBRE VARCHAR(80) NOT NULL,
  PRIMARY KEY(IDPAIS, IDPROVINCIA, IDCIUDAD),
  FOREIGN KEY(IDPAIS, IDPROVINCIA) REFERENCES TV_PROVINCIA(IDPAIS,
IDPROVINCIA)
)TYPE=INNODB;

CREATE TABLE TV_PERSONA(
  ID VARCHAR(13) NOT NULL,
  NOMBRES VARCHAR(80) NOT NULL,
  APELLIDOS VARCHAR(80) NOT NULL,
  CEDULA VARCHAR(13),
  DIRECCION VARCHAR(150),
  TELEFONO VARCHAR(20),
  FECHANACIMIENTO DATE,
  EMAIL VARCHAR(150) NOT NULL,
  CONTRASENA VARCHAR(30) NOT NULL,
  IDPAIS INT NOT NULL,
  IDPROVINCIA INT NOT NULL,
  IDCIUDAD INT NOT NULL,
  SEXO CHAR(1) NOT NULL,
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_CLIENTE(
  ID VARCHAR(13) NOT NULL,
  DIRECCIONFACTURACION VARCHAR(150),
  FAX VARCHAR(20),
  CELULAR VARCHAR(20),
  EMAILSECUNDARIO VARCHAR(150),
```

```
  BOLETIN CHAR(1),
  PRIMARY KEY(ID),
  FOREIGN KEY(ID) REFERENCES TV_PERSONA(ID)
)TYPE=INNODB;

CREATE TABLE TV_OPERADOR(
  ID VARCHAR(13) NOT NULL,
  TIPO VARCHAR(20) NOT NULL,
  PRIMARY KEY(ID),
  FOREIGN KEY(ID) REFERENCES TV_PERSONA(ID)
)TYPE=INNODB;

CREATE TABLE TV_PRODUCTO(
  ID VARCHAR(13) NOT NULL,
  TITULOES VARCHAR(100),
  TITULOEN VARCHAR(100),
  DESCRIPCIONES VARCHAR(512),
  DESCRIPCIONEN VARCHAR(512),
  PRECIO DOUBLE,
  EXISTENCIA INT,
  PESO DOUBLE,
  MOSTRARPORTADA CHAR(1),
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_PROVEEDOR(
  ID VARCHAR(13) NOT NULL,
  RUC VARCHAR(13),
  NOMBRE VARCHAR(150) NOT NULL,
  DIRECCION VARCHAR(150),
  TELEFONO VARCHAR(20),
  EMAIL VARCHAR(150) NOT NULL,
  PAIS VARCHAR(80),
  PROVINCIA VARCHAR(80),
  CIUDAD VARCHAR(80),
  CONTACTO VARCHAR(100) NOT NULL,
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_CATEGORIA(
  ID VARCHAR(13) NOT NULL,
  NOMBREES VARCHAR(100),
  NOMBREEN VARCHAR(100),
  ORDEN INT,
  IMAGENES VARCHAR(50),
  IMAGENEN VARCHAR(50),
  MOSTRAR CHAR(1),
  NIVELIDENTACION INT,
  SECCION CHAR(1),
```

```
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_MENSAJE(
  ID VARCHAR(13) NOT NULL,
  MFROM VARCHAR(13) NOT NULL,
  MTO VARCHAR(13) NOT NULL,
  TITULO VARCHAR(120),
  MENSAJE VARCHAR(1024),
  FECHA TIMESTAMP,
  LEIDO CHAR(1),
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_COMENTARIOPRODUCTO(
  ID VARCHAR(13) NOT NULL,
  TEXTO VARCHAR(512) NOT NULL,
  FECHA TIMESTAMP,
  NOMBRE VARCHAR(80) NOT NULL,
  EMAIL VARCHAR(100) NOT NULL,
  IDPRODUCTO VARCHAR(13),
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_CATEGORIAPRODUCTO(
  IDCATEGORIA VARCHAR(13),
  IDPRODUCTO VARCHAR(13),
  PRIMARY KEY(IDCATEGORIA, IDPRODUCTO)
)TYPE=INNODB;

CREATE TABLE TV_FOTO(
  ID VARCHAR(13),
  PATH VARCHAR(64),
  IDPRODUCTO VARCHAR(13),
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_PEDIDO(
  ID VARCHAR(13),
  FECHA DATETIME,
  TOTAL DOUBLE,
  FORMAPAGO VARCHAR(50),
  PAGADO CHAR(1),
  ENTREGADO CHAR(1),
  DIRECCIONENVIO VARCHAR(180),
  DIRECCIONFACTURACION VARCHAR(180),
  EMPRESATRANSPORTE VARCHAR(50),
  NROGUIA VARCHAR(150),
  IDCLIENTE VARCHAR(13),
```

```
  COSTOTRANSPORTE DOUBLE,
  NRODOCUMENTOPAGO VARCHAR(80),
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_DETALLEPEDIDO(
  ITEM INT,
  CANTIDAD INT,
  PRECIO DOUBLE,
  SUBTOTAL DOUBLE,
  COSTOENVIO DOUBLE,
  IDPEDIDO VARCHAR(13),
  IDPRODUCTO VARCHAR(13),
  PRIMARY KEY(IDPEDIDO, IDPRODUCTO)
)TYPE=INNODB;

CREATE TABLE TV_COMPRA(
  ID VARCHAR(13),
  FECHA DATETIME,
  TOTAL DOUBLE,
  IDPROVEEDOR VARCHAR(13),
  PRIMARY KEY(ID)
)TYPE=INNODB;

CREATE TABLE TV_DETALLECOMPRA(
  ITEM INT,
  CANTIDAD INT,
  PRECIO DOUBLE,
  SUBTOTAL DOUBLE,
  IDCOMPRA VARCHAR(13),
  IDPRODUCTO VARCHAR(13),
  PRIMARY KEY(IDCOMPRA, IDPRODUCTO)
)TYPE=INNODB;
```

7.3.2.2 Script de creación de inicio de datos

```
INSERT INTO TV_PAIS VALUES (1,'Afghanistan','AF','AFG','1');
INSERT INTO TV_PAIS VALUES (2,'Albania','AL','ALB','1');
INSERT INTO TV_PAIS VALUES (3,'Algeria','DZ','DZA','1');
INSERT INTO TV_PAIS VALUES (4,'American Samoa','AS','ASM','1');
INSERT INTO TV_PAIS VALUES (5,'Andorra','AD','AND','1');
INSERT INTO TV_PAIS VALUES (6,'Angola','AO','AGO','1');
INSERT INTO TV_PAIS VALUES (7,'Anguilla','AI','AIA','1');
INSERT INTO TV_PAIS VALUES (8,'Antarctica','AQ','ATA','1');
INSERT INTO TV_PAIS VALUES (9,'Antigua and Barbuda','AG','ATG','1');
INSERT INTO TV_PAIS VALUES (10,'Argentina','AR','ARG','1');
INSERT INTO TV_PAIS VALUES (11,'Armenia','AM','ARM','1');
INSERT INTO TV_PAIS VALUES (12,'Aruba','AW','ABW','1');
```

```
INSERT INTO TV_PAIS VALUES (13,'Australia','AU','AUS','1');
INSERT INTO TV_PAIS VALUES (14,'Austria','AT','AUT','5');
INSERT INTO TV_PAIS VALUES (15,'Azerbaijan','AZ','AZE','1');
INSERT INTO TV_PAIS VALUES (16,'Bahamas','BS','BHS','1');
INSERT INTO TV_PAIS VALUES (17,'Bahrain','BH','BHR','1');
INSERT INTO TV_PAIS VALUES (18,'Bangladesh','BD','BGD','1');
INSERT INTO TV_PAIS VALUES (19,'Barbados','BB','BRB','1');
INSERT INTO TV_PAIS VALUES (20,'Belarus','BY','BLR','1');
INSERT INTO TV_PAIS VALUES (21,'Belgium','BE','BEL','1');
INSERT INTO TV_PAIS VALUES (22,'Belize','BZ','BLZ','1');
INSERT INTO TV_PAIS VALUES (23,'Benin','BJ','BEN','1');
INSERT INTO TV_PAIS VALUES (24,'Bermuda','BM','BMU','1');
INSERT INTO TV_PAIS VALUES (25,'Bhutan','BT','BTN','1');
INSERT INTO TV_PAIS VALUES (26,'Bolivia','BO','BOL','1');
INSERT INTO TV_PAIS VALUES (27,'Bosnia and Herzegowina','BA','BIH','1');
INSERT INTO TV_PAIS VALUES (28,'Botswana','BW','BWA','1');
INSERT INTO TV_PAIS VALUES (29,'Bouvet Island','BV','BVT','1');
INSERT INTO TV_PAIS VALUES (30,'Brazil','BR','BRA','1');
INSERT INTO TV_PAIS VALUES (31,'British Indian Ocean Territory','IO','IOT','1');
INSERT INTO TV_PAIS VALUES (32,'Brunei Darussalam','BN','BRN','1');
INSERT INTO TV_PAIS VALUES (33,'Bulgaria','BG','BGR','1');
INSERT INTO TV_PAIS VALUES (34,'Burkina Faso','BF','BFA','1');
INSERT INTO TV_PAIS VALUES (35,'Burundi','BI','BDI','1');
INSERT INTO TV_PAIS VALUES (36,'Cambodia','KH','KHM','1');
INSERT INTO TV_PAIS VALUES (37,'Cameroon','CM','CMR','1');
INSERT INTO TV_PAIS VALUES (38,'Canada','CA','CAN','1');
INSERT INTO TV_PAIS VALUES (39,'Cape Verde','CV','CPV','1');
INSERT INTO TV_PAIS VALUES (40,'Cayman Islands','KY','CYM','1');
INSERT INTO TV_PAIS VALUES (41,'Central African Republic','CF','CAF','1');
INSERT INTO TV_PAIS VALUES (42,'Chad','TD','TCD','1');
INSERT INTO TV_PAIS VALUES (43,'Chile','CL','CHL','1');
INSERT INTO TV_PAIS VALUES (44,'China','CN','CHN','1');
INSERT INTO TV_PAIS VALUES (45,'Christmas Island','CX','CXR','1');
INSERT INTO TV_PAIS VALUES (46,'Cocos (Keeling) Islands','CC','CCK','1');
INSERT INTO TV_PAIS VALUES (47,'Colombia','CO','COL','1');
INSERT INTO TV_PAIS VALUES (48,'Comoros','KM','COM','1');
INSERT INTO TV_PAIS VALUES (49,'Congo','CG','COG','1');
INSERT INTO TV_PAIS VALUES (50,'Cook Islands','CK','COK','1');
INSERT INTO TV_PAIS VALUES (51,'Costa Rica','CR','CRI','1');
INSERT INTO TV_PAIS VALUES (52,'Cote D\'Ivoire','CI','CIV','1');
INSERT INTO TV_PAIS VALUES (53,'Croatia','HR','HRV','1');
INSERT INTO TV_PAIS VALUES (54,'Cuba','CU','CUB','1');
INSERT INTO TV_PAIS VALUES (55,'Cyprus','CY','CYP','1');
INSERT INTO TV_PAIS VALUES (56,'Czech Republic','CZ','CZE','1');
INSERT INTO TV_PAIS VALUES (57,'Denmark','DK','DNK','1');
INSERT INTO TV_PAIS VALUES (58,'Djibouti','DJ','DJI','1');
INSERT INTO TV_PAIS VALUES (59,'Dominica','DM','DMA','1');
INSERT INTO TV_PAIS VALUES (60,'Dominican Republic','DO','DOM','1');
INSERT INTO TV_PAIS VALUES (61,'East Timor','TP','TMP','1');
```

INSERT INTO TV_PAIS VALUES (62,'Ecuador','EC','ECU','1');
INSERT INTO TV_PAIS VALUES (63,'Egypt','EG','EGY','1');
INSERT INTO TV_PAIS VALUES (64,'El Salvador','SV','SLV','1');
INSERT INTO TV_PAIS VALUES (65,'Equatorial Guinea','GQ','GNQ','1');
INSERT INTO TV_PAIS VALUES (66,'Eritrea','ER','ERI','1');
INSERT INTO TV_PAIS VALUES (67,'Estonia','EE','EST','1');
INSERT INTO TV_PAIS VALUES (68,'Ethiopia','ET','ETH','1');
INSERT INTO TV_PAIS VALUES (69,'Falkland Islands (Malvinas)','FK','FLK','1');
INSERT INTO TV_PAIS VALUES (70,'Faroe Islands','FO','FRO','1');
INSERT INTO TV_PAIS VALUES (71,'Fiji','FJ','FJI','1');
INSERT INTO TV_PAIS VALUES (72,'Finland','FI','FIN','1');
INSERT INTO TV_PAIS VALUES (73,'France','FR','FRA','1');
INSERT INTO TV_PAIS VALUES (74,'France, Metropolitan','FX','FXX','1');
INSERT INTO TV_PAIS VALUES (75,'French Guiana','GF','GUF','1');
INSERT INTO TV_PAIS VALUES (76,'French Polynesia','PF','PYF','1');
INSERT INTO TV_PAIS VALUES (77,'French Southern Territories','TF','ATF','1');
INSERT INTO TV_PAIS VALUES (78,'Gabon','GA','GAB','1');
INSERT INTO TV_PAIS VALUES (79,'Gambia','GM','GMB','1');
INSERT INTO TV_PAIS VALUES (80,'Georgia','GE','GEO','1');
INSERT INTO TV_PAIS VALUES (81,'Germany','DE','DEU','5');
INSERT INTO TV_PAIS VALUES (82,'Ghana','GH','GHA','1');
INSERT INTO TV_PAIS VALUES (83,'Gibraltar','GI','GIB','1');
INSERT INTO TV_PAIS VALUES (84,'Greece','GR','GRC','1');
INSERT INTO TV_PAIS VALUES (85,'Greenland','GL','GRL','1');
INSERT INTO TV_PAIS VALUES (86,'Grenada','GD','GRD','1');
INSERT INTO TV_PAIS VALUES (87,'Guadeloupe','GP','GLP','1');
INSERT INTO TV_PAIS VALUES (88,'Guam','GU','GUM','1');
INSERT INTO TV_PAIS VALUES (89,'Guatemala','GT','GTM','1');
INSERT INTO TV_PAIS VALUES (90,'Guinea','GN','GIN','1');
INSERT INTO TV_PAIS VALUES (91,'Guinea-bissau','GW','GNB','1');
INSERT INTO TV_PAIS VALUES (92,'Guyana','GY','GUY','1');
INSERT INTO TV_PAIS VALUES (93,'Haiti','HT','HTI','1');
INSERT INTO TV_PAIS VALUES (94,'Heard and Mc Donald Islands','HM','HMD','1');
INSERT INTO TV_PAIS VALUES (95,'Honduras','HN','HND','1');
INSERT INTO TV_PAIS VALUES (96,'Hong Kong','HK','HKG','1');
INSERT INTO TV_PAIS VALUES (97,'Hungary','HU','HUN','1');
INSERT INTO TV_PAIS VALUES (98,'Iceland','IS','ISL','1');
INSERT INTO TV_PAIS VALUES (99,'India','IN','IND','1');
INSERT INTO TV_PAIS VALUES (100,'Indonesia','ID','IDN','1');
INSERT INTO TV_PAIS VALUES (101,'Iran (Islamic Republic of)','IR','IRN','1');
INSERT INTO TV_PAIS VALUES (102,'Iraq','IQ','IRQ','1');
INSERT INTO TV_PAIS VALUES (103,'Ireland','IE','IRL','1');
INSERT INTO TV_PAIS VALUES (104,'Israel','IL','ISR','1');
INSERT INTO TV_PAIS VALUES (105,'Italy','IT','ITA','1');
INSERT INTO TV_PAIS VALUES (106,'Jamaica','JM','JAM','1');
INSERT INTO TV_PAIS VALUES (107,'Japan','JP','JPN','1');
INSERT INTO TV_PAIS VALUES (108,'Jordan','JO','JOR','1');
INSERT INTO TV_PAIS VALUES (109,'Kazakhstan','KZ','KAZ','1');
INSERT INTO TV_PAIS VALUES (110,'Kenya','KE','KEN','1');

INSERT INTO TV_PAIS VALUES (111,'Kiribati','KI','KIR','1');
INSERT INTO TV_PAIS VALUES (112,'Korea, Democratic People\'s Republic of','KP','PRK','1');
INSERT INTO TV_PAIS VALUES (113,'Korea, Republic of','KR','KOR','1');
INSERT INTO TV_PAIS VALUES (114,'Kuwait','KW','KWT','1');
INSERT INTO TV_PAIS VALUES (115,'Kyrgyzstan','KG','KGZ','1');
INSERT INTO TV_PAIS VALUES (116,'Lao People\'s Democratic Republic','LA','LAO','1');
INSERT INTO TV_PAIS VALUES (117,'Latvia','LV','LVA','1');
INSERT INTO TV_PAIS VALUES (118,'Lebanon','LB','LBN','1');
INSERT INTO TV_PAIS VALUES (119,'Lesotho','LS','LSO','1');
INSERT INTO TV_PAIS VALUES (120,'Liberia','LR','LBR','1');
INSERT INTO TV_PAIS VALUES (121,'Libyan Arab Jamahiriya','LY','LBY','1');
INSERT INTO TV_PAIS VALUES (122,'Liechtenstein','LI','LIE','1');
INSERT INTO TV_PAIS VALUES (123,'Lithuania','LT','LTU','1');
INSERT INTO TV_PAIS VALUES (124,'Luxembourg','LU','LUX','1');
INSERT INTO TV_PAIS VALUES (125,'Macau','MO','MAC','1');
INSERT INTO TV_PAIS VALUES (126,'Macedonia, The Former Yugoslav Republic of','MK','MKD','1');
INSERT INTO TV_PAIS VALUES (127,'Madagascar','MG','MDG','1');
INSERT INTO TV_PAIS VALUES (128,'Malawi','MW','MWI','1');
INSERT INTO TV_PAIS VALUES (129,'Malaysia','MY','MYS','1');
INSERT INTO TV_PAIS VALUES (130,'Maldives','MV','MDV','1');
INSERT INTO TV_PAIS VALUES (131,'Mali','ML','MLI','1');
INSERT INTO TV_PAIS VALUES (132,'Malta','MT','MLT','1');
INSERT INTO TV_PAIS VALUES (133,'Marshall Islands','MH','MHL','1');
INSERT INTO TV_PAIS VALUES (134,'Martinique','MQ','MTQ','1');
INSERT INTO TV_PAIS VALUES (135,'Mauritania','MR','MRT','1');
INSERT INTO TV_PAIS VALUES (136,'Mauritius','MU','MUS','1');
INSERT INTO TV_PAIS VALUES (137,'Mayotte','YT','MYT','1');
INSERT INTO TV_PAIS VALUES (138,'Mexico','MX','MEX','1');
INSERT INTO TV_PAIS VALUES (139,'Micronesia, Federated States of','FM','FSM','1');
INSERT INTO TV_PAIS VALUES (140,'Moldova, Republic of','MD','MDA','1');
INSERT INTO TV_PAIS VALUES (141,'Monaco','MC','MCO','1');
INSERT INTO TV_PAIS VALUES (142,'Mongolia','MN','MNG','1');
INSERT INTO TV_PAIS VALUES (143,'Montserrat','MS','MSR','1');
INSERT INTO TV_PAIS VALUES (144,'Morocco','MA','MAR','1');
INSERT INTO TV_PAIS VALUES (145,'Mozambique','MZ','MOZ','1');
INSERT INTO TV_PAIS VALUES (146,'Myanmar','MM','MMR','1');
INSERT INTO TV_PAIS VALUES (147,'Namibia','NA','NAM','1');
INSERT INTO TV_PAIS VALUES (148,'Nauru','NR','NRU','1');
INSERT INTO TV_PAIS VALUES (149,'Nepal','NP','NPL','1');
INSERT INTO TV_PAIS VALUES (150,'Netherlands','NL','NLD','1');
INSERT INTO TV_PAIS VALUES (151,'Netherlands Antilles','AN','ANT','1');
INSERT INTO TV_PAIS VALUES (152,'New Caledonia','NC','NCL','1');
INSERT INTO TV_PAIS VALUES (153,'New Zealand','NZ','NZL','1');
INSERT INTO TV_PAIS VALUES (154,'Nicaragua','NI','NIC','1');
INSERT INTO TV_PAIS VALUES (155,'Niger','NE','NER','1');
INSERT INTO TV_PAIS VALUES (156,'Nigeria','NG','NGA','1');
INSERT INTO TV_PAIS VALUES (157,'Niue','NU','NIU','1');

INSERT INTO TV_PAIS VALUES (158,'Norfolk Island','NF','NFK','1');
INSERT INTO TV_PAIS VALUES (159,'Northern Mariana Islands','MP','MNP','1');
INSERT INTO TV_PAIS VALUES (160,'Norway','NO','NOR','1');
INSERT INTO TV_PAIS VALUES (161,'Oman','OM','OMN','1');
INSERT INTO TV_PAIS VALUES (162,'Pakistan','PK','PAK','1');
INSERT INTO TV_PAIS VALUES (163,'Palau','PW','PLW','1');
INSERT INTO TV_PAIS VALUES (164,'Panama','PA','PAN','1');
INSERT INTO TV_PAIS VALUES (165,'Papua New Guinea','PG','PNG','1');
INSERT INTO TV_PAIS VALUES (166,'Paraguay','PY','PRY','1');
INSERT INTO TV_PAIS VALUES (167,'Peru','PE','PER','1');
INSERT INTO TV_PAIS VALUES (168,'Philippines','PH','PHL','1');
INSERT INTO TV_PAIS VALUES (169,'Pitcairn','PN','PCN','1');
INSERT INTO TV_PAIS VALUES (170,'Poland','PL','POL','1');
INSERT INTO TV_PAIS VALUES (171,'Portugal','PT','PRT','1');
INSERT INTO TV_PAIS VALUES (172,'Puerto Rico','PR','PRI','1');
INSERT INTO TV_PAIS VALUES (173,'Qatar','QA','QAT','1');
INSERT INTO TV_PAIS VALUES (174,'Reunion','RE','REU','1');
INSERT INTO TV_PAIS VALUES (175,'Romania','RO','ROM','1');
INSERT INTO TV_PAIS VALUES (176,'Russian Federation','RU','RUS','1');
INSERT INTO TV_PAIS VALUES (177,'Rwanda','RW','RWA','1');
INSERT INTO TV_PAIS VALUES (178,'Saint Kitts and Nevis','KN','KNA','1');
INSERT INTO TV_PAIS VALUES (179,'Saint Lucia','LC','LCA','1');
INSERT INTO TV_PAIS VALUES (180,'Saint Vincent and the Grenadines','VC','VCT','1');
INSERT INTO TV_PAIS VALUES (181,'Samoa','WS','WSM','1');
INSERT INTO TV_PAIS VALUES (182,'San Marino','SM','SMR','1');
INSERT INTO TV_PAIS VALUES (183,'Sao Tome and Principe','ST','STP','1');
INSERT INTO TV_PAIS VALUES (184,'Saudi Arabia','SA','SAU','1');
INSERT INTO TV_PAIS VALUES (185,'Senegal','SN','SEN','1');
INSERT INTO TV_PAIS VALUES (186,'Seychelles','SC','SYC','1');
INSERT INTO TV_PAIS VALUES (187,'Sierra Leone','SL','SLE','1');
INSERT INTO TV_PAIS VALUES (188,'Singapore','SG','SGP', '4');
INSERT INTO TV_PAIS VALUES (189,'Slovakia (Slovak Republic)','SK','SVK','1');
INSERT INTO TV_PAIS VALUES (190,'Slovenia','SI','SVN','1');
INSERT INTO TV_PAIS VALUES (191,'Solomon Islands','SB','SLB','1');
INSERT INTO TV_PAIS VALUES (192,'Somalia','SO','SOM','1');
INSERT INTO TV_PAIS VALUES (193,'South Africa','ZA','ZAF','1');
INSERT INTO TV_PAIS VALUES (194,'South Georgia and the South Sandwich Islands','GS','SGS','1');
INSERT INTO TV_PAIS VALUES (195,'Spain','ES','ESP','3');
INSERT INTO TV_PAIS VALUES (196,'Sri Lanka','LK','LKA','1');
INSERT INTO TV_PAIS VALUES (197,'St. Helena','SH','SHN','1');
INSERT INTO TV_PAIS VALUES (198,'St. Pierre and Miquelon','PM','SPM','1');
INSERT INTO TV_PAIS VALUES (199,'Sudan','SD','SDN','1');
INSERT INTO TV_PAIS VALUES (200,'Suriname','SR','SUR','1');
INSERT INTO TV_PAIS VALUES (201,'Svalbard and Jan Mayen Islands','SJ','SJM','1');
INSERT INTO TV_PAIS VALUES (202,'Swaziland','SZ','SWZ','1');
INSERT INTO TV_PAIS VALUES (203,'Sweden','SE','SWE','1');
INSERT INTO TV_PAIS VALUES (204,'Switzerland','CH','CHE','1');
INSERT INTO TV_PAIS VALUES (205,'Syrian Arab Republic','SY','SYR','1');

INSERT INTO TV_PAIS VALUES (206,'Taiwan','TW','TWN','1');
INSERT INTO TV_PAIS VALUES (207,'Tajikistan','TJ','TJK','1');
INSERT INTO TV_PAIS VALUES (208,'Tanzania, United Republic of','TZ','TZA','1');
INSERT INTO TV_PAIS VALUES (209,'Thailand','TH','THA','1');
INSERT INTO TV_PAIS VALUES (210,'Togo','TG','TGO','1');
INSERT INTO TV_PAIS VALUES (211,'Tokelau','TK','TKL','1');
INSERT INTO TV_PAIS VALUES (212,'Tonga','TO','TON','1');
INSERT INTO TV_PAIS VALUES (213,'Trinidad and Tobago','TT','TTO','1');
INSERT INTO TV_PAIS VALUES (214,'Tunisia','TN','TUN','1');
INSERT INTO TV_PAIS VALUES (215,'Turkey','TR','TUR','1');
INSERT INTO TV_PAIS VALUES (216,'Turkmenistan','TM','TKM','1');
INSERT INTO TV_PAIS VALUES (217,'Turks and Caicos Islands','TC','TCA','1');
INSERT INTO TV_PAIS VALUES (218,'Tuvalu','TV','TUV','1');
INSERT INTO TV_PAIS VALUES (219,'Uganda','UG','UGA','1');
INSERT INTO TV_PAIS VALUES (220,'Ukraine','UA','UKR','1');
INSERT INTO TV_PAIS VALUES (221,'United Arab Emirates','AE','ARE','1');
INSERT INTO TV_PAIS VALUES (222,'United Kingdom','GB','GBR','1');
INSERT INTO TV_PAIS VALUES (223,'United States','US','USA', '2');
INSERT INTO TV_PAIS VALUES (224,'United States Minor Outlying Islands','UM','UMI','1');
INSERT INTO TV_PAIS VALUES (225,'Uruguay','UY','URY','1');
INSERT INTO TV_PAIS VALUES (226,'Uzbekistan','UZ','UZB','1');
INSERT INTO TV_PAIS VALUES (227,'Vanuatu','VU','VUT','1');
INSERT INTO TV_PAIS VALUES (228,'Vatican City State (Holy See)','VA','VAT','1');
INSERT INTO TV_PAIS VALUES (229,'Venezuela','VE','VEN','1');
INSERT INTO TV_PAIS VALUES (230,'Viet Nam','VN','VNM','1');
INSERT INTO TV_PAIS VALUES (231,'Virgin Islands (British)','VG','VGB','1');
INSERT INTO TV_PAIS VALUES (232,'Virgin Islands (U.S.)','VI','VIR','1');
INSERT INTO TV_PAIS VALUES (233,'Wallis and Futuna Islands','WF','WLF','1');
INSERT INTO TV_PAIS VALUES (234,'Western Sahara','EH','ESH','1');
INSERT INTO TV_PAIS VALUES (235,'Yemen','YE','YEM','1');
INSERT INTO TV_PAIS VALUES (236,'Yugoslavia','YU','YUG','1');
INSERT INTO TV_PAIS VALUES (237,'Zaire','ZR','ZAR','1');
INSERT INTO TV_PAIS VALUES (238,'Zambia','ZM','ZMB','1');
INSERT INTO TV_PAIS VALUES (239,'Zimbabwe','ZW','ZWE','1');

INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 1, 'Azuay');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 2, 'BolÃ-var');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 3, 'CaÃ±ar');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 4, 'Carchi');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 5, 'Chimborazo');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 6, 'Cotopaxi');

INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 7, 'El Oro');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 8, 'Esmeraldas');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 9, 'GalÃ¡pagos');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 10, 'Guayas');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 11, 'Imbabura');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 12, 'Loja');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 13, 'Los RÃos');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 14, 'ManabÃ');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 15, 'Morona Santiago');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 16, 'Napo');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 17, 'Orellana');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 18, 'Pastaza');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 19, 'Pichincha');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 20, 'Santa Elena');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 21, 'Sto. Domingo de los TsÃ¡chilas');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 22, 'SucumbÃos');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 23, 'Tungurahua');
INSERT INTO TV_PROVINCIA (IDPAIS, IDPROVINCIA, NOMBRE) VALUES(62, 24, 'Zamora Chinchipe');

7.3.4 Archivos de configuración

7.3.4.1 Archivo de configuración de la tienda virtual en Tomcal web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
```

```
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
   version="2.5">

    <description>
      Carrito de compras.
    </description>
    <display-name>Servlet and JSP Examples</display-name>

    <!-- Define servlets that are included in the example application -->

    <servlet>
        <servlet-name>ProductoInfo</servlet-name>
        <servlet-class>ProductoInfo</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>ProductoInfo</servlet-name>
        <url-pattern>/ProductoInfo</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>ProveedorInfo</servlet-name>
        <servlet-class>ProveedorInfo</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>ProveedorInfo</servlet-name>
        <url-pattern>/ProveedorInfo</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>CategoriaInfo</servlet-name>
        <servlet-class>CategoriaInfo</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>CategoriaInfo</servlet-name>
```

```
    <url-pattern>/CategoriaInfo</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>ClienteInfo</servlet-name>
    <servlet-class>ClienteInfo</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>ClienteInfo</servlet-name>
    <url-pattern>/ClienteInfo</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>Browse</servlet-name>
    <servlet-class>Browse</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Browse</servlet-name>
    <url-pattern>/Browse</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>RegionInfo</servlet-name>
    <servlet-class>RegionInfo</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>RegionInfo</servlet-name>
    <url-pattern>/RegionInfo</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>Cart</servlet-name>
    <servlet-class>Cart</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Cart</servlet-name>
    <url-pattern>/Cart</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>CompraInfo</servlet-name>
    <servlet-class>CompraInfo</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>CompraInfo</servlet-name>
    <url-pattern>/CompraInfo</url-pattern>
</servlet-mapping>

<servlet>
```

```
  <servlet-name>UploadFichero</servlet-name>
  <servlet-class>UploadFichero</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>UploadFichero</servlet-name>
    <url-pattern>/UploadFichero</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>AdministrarClientes</servlet-name>
  <servlet-class>AdministrarClientes</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>AdministrarClientes</servlet-name>
    <url-pattern>/AdministrarClientes</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>Administracion</servlet-name>
  <servlet-class>Administracion</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Administracion</servlet-name>
    <url-pattern>/Administracion</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>OperadorInfo</servlet-name>
  <servlet-class>OperadorInfo</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>OperadorInfo</servlet-name>
    <url-pattern>/OperadorInfo</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>MensajeInfo</servlet-name>
  <servlet-class>MensajeInfo</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>MensajeInfo</servlet-name>
    <url-pattern>/MensajeInfo</url-pattern>
</servlet-mapping>
```

```
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
</welcome-file-list>

        <security-constraint>
          <web-resource-collection>
            <web-resource-name>Entire  Application</web-resource-name>
            <url-pattern>/secure/*</url-pattern>
                  <http-method>GET</http-method>
                  <http-method>POST</http-method>

          </web-resource-collection>

          <user-data-constraint>
            <transport-guarantee>CONFIDENTIAL</transport-guarantee>
          </user-data-constraint>
        </security-constraint>

        <security-constraint>
          <web-resource-collection>
            <web-resource-name>Entire  Application</web-resource-name>
            <url-pattern>/client/*</url-pattern>
                  <http-method>GET</http-method>
                  <http-method>POST</http-method>

          </web-resource-collection>

          <user-data-constraint>
            <transport-guarantee>CONFIDENTIAL</transport-guarantee>
          </user-data-constraint>
        </security-constraint>

</web-app>
```

7.4 Pruebas

Se procedió a realizar pruebas del correcto funcionamiento de la tienda virtual, dando énfasis a la utilización de diferentes plataformas de software.

En lo referente al servidor, se realizaron pruebas sobre los sistemas operativos GNU Linux distribución Red Hat y Ubuntu, y el servidor Tomcat versiones 5.5 y 6.0. En lo referente a servicios Web se utilizó la plataforma AXIS 1.4 para las pruebas de funcionamiento.

En cuanto a la utilización por parte del cliente se realizaron pruebas sobre los sistemas operativos GNU Linux distribución Ubuntu 10.04.1 y Windows Vista, el navegador Firefox 3.0.19 e Internet Explorer 7.

De los resultados de las pruebas se puede destacar que existen funcionalidades de JavaScript que trabajan bien sobre el navegador Firefox y que no funcionan sobre Internet Explorer, por lo que se debe corregir utilizando procedimientos estándar que sirvan para ambos navegadores. Así mismo es destacable que ciertos tipos de letra utilizados por las hojas de estilo de cascada no producían los mismos resultados en ambos navegadores.

Por otra parte, se encontró falencias en el despliegue de las tildes y de la letra eñe, puesto que se utilizan distintas codificaciones de caracteres, siendo las más importantes la Occidental (ISO-8859-1) y Unicode (UTF-8), existen otras codificaciones para otros idiomas.

En lo referente al contenedor Web Tomcat en sus versiones 5.5 y 6.0 se encontró que la tecnología JavaBeans presenta algunas diferencias en estas versiones, por lo que se procedió a buscar métodos estándar para estas versiones.

En lo referente a la seguridad, se encontró que no funcionaba el certificado digital para utilizar el protocolo SSL/TLS, y se nos recomendó la utilización de una IP dedicada para poder hacer funcionar esta característica.

3.5 Implantación

La implantación de la tienda virtual se la realizó alquilando el hosting y comprando un dominio para la misma. El servidor utiliza como sistema operativo GNU Linux distribución Red Hat, el contenedor Web Apache Tomcat 5.5, la plataforma AXIS para servicios Web y el gestor de base de datos MySQL 5.1

La URL de la tienda virtual es www.jgexclusividades.com

Para la implementación de certificados digitales se adquirió la IP dedicada 76.73.86.109, con la que se puede implementar la seguridad mediante el protocolo SSL/TLS.

7.5.1 Pantallas de la tienda virtual

Luego de desarrollar el proyecto, procedemos a mostrar las pantallas más representativas de la tienda virtual.

7.5.1.1 Pantalla de inicio de la tienda virtual

Figura 7.17 Pantalla de inicio de la tienda virtual

7.5.1.2 Pantalla de muestra de productos

Figura 7.18 Pantalla de muestra de productos

7.5.1.3 Pantalla del Carrito de Compras

Figura 7.19 Pantalla del Carrito de Compras

7.5.1.4 Pantalla de la cuenta del cliente

Figura 7.20 Pantalla de la cuenta del cliente

7.5.1.5 Pantalla de pedidos de un cliente

Figura 7.21 Pantalla de pedidos de un cliente

7.5.1.6 Pantalla de administración de productos

Figura 7.22 Pantalla de administración de productos

7.5.1.7 Pantalla de administración de categorías de produtos

Figura 7.23 Pantalla de administración de categorías de productos

7.5.1.8 Pantalla de administración de proveedores

Figura 7.24 Pantalla de administración de proveedores

7.5.1.9 Pantalla de administración de pedidos

Figura 7.25 Pantalla de administración de pedidos

7.5.1.10 Pantalla de administración de compras

Figura 7.26 Pantalla de administración de compras

7.5.1.11 Pantalla de administración de operadores

Figura 7.27 Pantalla de administración de operadores

Conclusiones y recomendaciones

- La construcción de tiendas virtuales de Comercio Electrónico posibilitan innumerables beneficios para las empresas como el acceso a mercados mucho mayores, reducción de costos de promoción y distribución, ahorro por concepto de contratación de vendedores que promocionen los productos en diferentes lugares, etc.

- El Comercio Electrónico C2C (Consumer to Consumer), ha tenido un desarrollo muy importante, y está relacionado principalmente a sitios de subastas en Internet. La compra/venta de productos nuevos y usados por parte de consumidores a otros consumidores residentes en diferentes locaciones ha permitido su creciente popularidad.

- Los aspectos de seguridad de la información cumplen un rol vital en las transacciones realizadas sobre las redes de computadoras abiertas e inseguras como es el caso del Internet. Las transacciones de tipo comercial o financiera que se llevan a cabo en el Internet incluyen información sensible del cliente o consumidor, por tanto se evidencia la relevancia en proteger esta información.

- Cuando se incrementa la seguridad de un sistema informático, también aumenta el costo de brindar esa seguridad. Es relevante encontrar el punto de equilibrio entre costo vs. Seguridad. Se recomienda gastar en seguridad un monto inferior al costo de los bienes protegidos.

- Las aplicaciones de Comercio Electrónico y las aplicaciones financieras necesitan de la utilización de certificados digitales firmados por una Autoridad de Certificación, con lo que se consigue garantizar la seguridad de las mismas. Los certificados digitales auto firmados son útiles en aplicaciones de tipo académico o aquellas en las que no se arriesgue información financiera.

- Los aspectos regulatorios de un país influyen en la disponibilidad de emprendimientos de Comercio Electrónico, puesto que se evidencia la necesidad de leyes para control de fraude y protección para el consumidor final.

- Uno de los grandes beneficios del Comercio Electrónico tiene que ver con la posibilidad de conseguir productos que no están disponibles en el medio, pero que pueden ser obtenidos fácilmente a través de Internet.

- La disponibilidad de acceso a Internet y los costos de acceso son factores que influyen en la utilización del Comercio Electrónico en un país, por lo que se puede prever que el mejoramiento de la infraestructura de telecomunicaciones facilitará el desarrollo del Comercio Electrónico.

- Existen áreas en las que el comercio electrónico es más fácil y viable de conducir, como es el caso del sector inmobiliario, sector hotelero, la reservación y compra de tickets aéreos, venta de contenido digital, etc.

- El sector laboral se ha visto beneficiado enormemente del uso de las TICs, resulta muy conveniente contratar servicios especializados como programadores, diseñadores, servicios de traducción, articulistas especializados, etc., la distribución del contenido digital se lo realiza fácilmente a través de Internet. Por su parte, la búsqueda y oferta de empleo a través de Internet se ha incrementado gracias a las bolsas de trabajo en Internet.

- El dinero digital es una tecnología prometedora que pudiera a futuro cambiar la forma en que se realizan los pagos por pequeños montos. El uso de tarjetas inteligentes facilita este tipo de pagos, puesto que permiten guardar la información de los clientes, saldos, etc. Otra posibilidad es el uso del teléfono móvil para realizar micro pagos, resulta muy conveniente y cómodo, sin embargo su desarrollo se encuentra apenas en sus inicios.

- Existen sitios como www.paypal.com, en los que se pueden realizar variados tipos de transacciones que facilitan la realización de pagos como envío de dinero, pagos a través de tarjetas de crédito, transferencias, etc.

- El m-commerce es el siguiente paso después del e-commerce. El Comercio Móvil facilita las transacciones del usuario, debido a que pueden ser realizadas en cualquier momento y en cualquier lugar. Una de las ventajas sobresalientes del m-commerce es su alta penetración en la población, y su parte más débil tiene que ver con la seguridad de la información. Los mecanismos de seguridad empleados en teléfonos móviles no deben ser computacionalmente costosos debido a las capacidades limitadas de estos dispositivos.

- AJAX es una filosofía de desarrollo que combina diferentes tecnologías para construir aplicaciones Web dinámicas, interactivas y más eficientes.

- Las aplicaciones Web son un tipo de aplicaciones que se pueden ejecutar desde cualquier lugar mediante un navegador Web. Son utilizadas por

muchos usuarios a través del Internet, permiten la reducción de costos de distribución y mantenimiento de sistemas. Finalmente, debido a que las especificaciones de los teléfonos móviles están mejorando, es factible acceder a las aplicaciones Web a través de estos dispositivos.

- Los Servicios Web representan una tecnología creciente y realmente útil. Ayudan a la interacción de aplicaciones Web, incluso si han sido desarrolladas sobre plataformas distintas. Los servicios Web tienen un gran potencial al existir directorios de servicios Web con funciones variadas que permiten la reutilización de código y disminuir considerablemente costos y tiempos de desarrollo.

- OpenSSL es una herramienta Open Source muy poderosa en cuestión de Criptografía, permite realizar encriptación de archivos, firmar digitalmente documentos, creación de certificados digitales auto-firmados, etc. Su fortaleza se basa en la inclusión de muchas funciones de algoritmos tanto simétricos como asimétricos.

- Existen maneras de disminuir el tiempo de desarrollo de aplicaciones Web. Están disponibles en Internet sitios en donde se pueden conseguir templates o plantillas Web, que contienen la estructura de una página web, ahorrando al desarrollador el diseño, la combinación de colores, etc. El trabajo del desarrollador, para este caso, consiste en la personalización de estos templates.

- Se recomienda el uso de la Web 2.0 para la promoción de los sitios Web en general: redes sociales, promoción mediante vídeos, etc.

- No existe la seguridad perfecta de la información. Sin embargo se puede emplear mecanismo de seguridad de tal modo que el ataque de un usuario malintencionado sea neutralizado.

Bibliografía y Fuentes de Información

Trabajos citados

2createawebsite.com. 2010. http://www.2createawebsite.com/money/build-online-store.html (último acceso: 05 de 01 de 2010).

Comercio Electronico Global. *Comercio Electronico Global.* 20 de 06 de 2010. http://www.e-global.es/ (último acceso: 15 de 07 de 2010).

CONATEL. *Consejo Nacional de Telecomunicaciones.* 03 de 2010. http://www.conatel.gov.ec/site_conatel/index.php?option=com_content&view=category&id=41&Itemid=167 (último acceso: 04 de 2010).

Cox, Mark J., Ralf S. Engelschall, Dr. Stephen Henson, y Ben Laurie. *OpenSSL.* http://www.openssl.org (último acceso: 09 de 04 de 2010).

Darin, Mg. Lic. Susana. *M- Commerce: compras con el Celular.*

Forné, Jordi. *Seguridad en Internet.* Barcelona.

García, Carlos. *Adictos al trabajo.* 07 de 04 de 2006. http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=axis (último acceso: 28 de 06 de 2010).

GNU. *GNU.* http://www.gnu.org/ (último acceso: 12 de 05 de 2010).

González, Carlos. *Comportamiento del consumidor y CRM.* San José, 2009.

González, Carlos. *Introducción al E-Business.* San José, 2009.

Hermanos Carreño. *Programación en castellano: Servlets y JSP.* 2010. http://www.programacion.com/articulo/servlets_y_jsp_82 (último acceso: 18 de 02 de 2010).

Holden, Greg. *1000 Best eBay Success Secrets.* Naperville, Illinois: Sourcebooks Inc., 2007.

Martínez, Fernando. *El ABC de la Criptografía y sus Algoritmos.*

NetBeans. *NetBeans.* 2010. http://netbeans.org/ (último acceso: 24 de 06 de 2010).

NTT DoCoMo. *Security Basics.* Tokyo, 2005.

Open Source Community. *Open Source.* 2009. http://www.opensource.org/ (último acceso: 12 de 05 de 2010).

Oracle. *Sun.* http://www.sun.com (último acceso: 12 de 11 de 2009).

OSCommerce. *OSCommerce.* http://www.oscommerce.com (último acceso: 12 de 12 de 2009).

Serrano, Carlos. *Ciberconta.* 2010. http://ciberconta.unizar.es/LECCION/eCONTA/INICIO.HTML (último acceso: 11 de 02 de 2010).

Supertel. *SUPERINTENDENCIA DE TELECOMUNICACIONES.* 02 de 12 de 2009. http://www.supertel.gov.ec/index.php/Ultimas-noticias/datos-de-cuentas-y-usuarios-de-internet-hasta-septiembre-de-2009.html (último acceso: 14 de 08 de 2009).

Symteco. http://www.symteco.com/mobile-payment-systems (último acceso: 28 de 06 de 2010).

The Apache Software Foundation. *Apache Tomcat.* 2010. http://tomcat.apache.org/ (último acceso: 05 de 2010).

Ubuntu Community. *Ubuntu.* 2010. http://www.ubuntu.com/ (último acceso: 23 de 04 de 2010).

W3 Consortium. *W3C.* 06 de 05 de 2010. http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb (último acceso: 12 de 06 de 2010).

w3schools. http://www.w3schools.com (último acceso: 28 de 05 de 2010).

Widegren, Lena. *Ericsson.* 2006. http://www.ericsson.com/ericsson/corpinfo/publications/telecomreport/archive/2006/january/valista.shtml (último acceso: 26 de 06 de 2010).

Wikipedia. «Wikipedia.» *Wikipedia.* http://es.wikipedia.org/wiki/Comercio_electronico (último acceso: 01 de 04 de 2010).

Índice de gráficos

Índice de tablas

Glosario de términos y abreviaturas

ADSL: Asyncronous Digital Suscriber Line (Línea de abonado digital asíncrona)

AES: Advanced Encryption Standard (Estándar de Encriptación Avanzado).

AJAX: Asyncronous JavaScript and XML (JavaScript Asíncrono y XML).

API: Application Programming Interface (Interfaz de Programación de Aplicaciones).

AR: Aumented Reality (Realidad Aumentada).

ARPAnet: Advanced Research Projects Agency Network (Red de la Agencia de Proyectos de Investigación Avanzados)

ASP: Active Server Pages (Páginas Activas de Servidor)

B2B: Business to Business (Negocio a Negocio).

B2C: Business to Customer (Negocio a Cliente).

B2G: Business to Government (Negocio a Gobierno).

C2B: Customer to Business (Cliente a Negocio).

C2C: Customer to Customer (Cliente a Cliente).

C2G: Citizen to Government (Ciudadano a Gobierno).

CA: Certification Authority (Autoridad de Certificación).

CGI: Common Gateway Interface (Interface de Pasarela Común).

cHTML: compact HTML (HTML compacto)

CRM: Customer Relashionship Management

CSS: Cascade Style Sheets (Hojas de estilo de cascada).

DAO: Data Access Object (Objeto de Acceso a Datos).

DES: Data Encryption Standard (Estándar de Encriptación de Datos).

DHTML: Dynamic HTML (HTML dinámico).

DOM: Document Object Model (Modelo de Objetos de Documento).

DoS: Denial of Service (Negación de Servicio).

DTD: Document Type Definition (Definición de Tipo de Documento).

FAQ: Frequently Asked Question (Preguntas más frecuentemente).

FSF: Free Software Foundation (Fundación de Software libre)

G2B: Government to Business (Gobierno a Negocio).

G2C: Government to Citizen (Gobierno a Ciudadano).

G2G: Government to Governmet (Gobierno a Gobierno).

GNU: GNU's Not Unix (GNU no es Unix).

GPL: General Public License (Licencia Pública General).

GUI: Graphics User Interface (Interfaz Gráfica de Usuario).

HTML: Hyper-Text Markup Language (Lenguaje de Marcado de Hipertexto)

HTTP: Hyper-Text Transfer Protocol (Protocolo de Transferencia de Hipertexto).

HTTPS: Hyper-Text Transfer Protocol Secure (Protocolo de Transferencia de Hipertexto seguro).

IDE: Integrated Development Environment (Ambiente de desarrollo integrado).

IDEA: International Data Encryption Algorithm (Algoritmo de Encriptación de Datos Internacional).

IETF: Internet Engineering Task Force (Fuerza de Trabajo de Ingeniería en Internet).

IP: Internet Protocol (Protocolo de Internet).

J2EE: Java 2 Enterprise Edition (Edición Empresarial 2 de Java).

JAX-WS: Java API para XML Web Services (API de Java para Servicios Web y XML).

JCA: Java Criptography Architecture (Arquitectura de Criptografía de Java).

JCP: Java Community Process (Proceso de la Comunidad Java).

JSON: JavaScript Object Notation (Notación de Objeto de JavaScript).

JSP: Java Server Pages (Páginas de Servidor de Java).

JSTL: Java Server Pages Standard Tag Library (Librería de etiquetas estándar de JSP)

MAC: Message Authentication Code (Código de Autenticación de Mensaje).

MD5: Message Digest 5.

MVC: Modelo Vista Controlador

NCSA: National Center for Supercomputing Application (Centro Nacional para Aplicaciones de Supercomputación).

NFC: Near Field Communication (Comunicación de campo cercano)

NIC: Network Interface Card (Tarjeta de Interfaz de Red).

OSI: Open Standard Interconnection (Interconexión Estandar Abierta).

PDA: Personal Digital Assitant (Asistente personal digital).

PGP: Pretty Good Privacy (Muy Buena Privacidad).

PHP: Hypertext Preprocessor (Preprocesador de Hipertexto)

PKI: Public Key Infrastructure (Infraestructura de clave pública).

RA: Registration Authority (Autoridad de Registro).

RFC: Request For Comment (Petición para comentario)

RPC: Remote Procedure Call (Llama de Procedimiento Remoto)

RSA: Rivest Shamir Adleman.

S/MIME: Secure/Multipurpose Internet Mail Extensions (Extensiones multipropósito de Mail e Internet .)

SEO: Search Engine Optimization (Optimización de motores de búsqueda).

SHA: Secure Hash Algorithm (Algortimo Hash Seguro)

SMS: Short-Message System (Sistema de mensajes cortos)

SOAP: Simple Object Access Control (Control Simple de Acceso a Objetos).

SQL: Structured Query Language (Lenguaje de Consultas Estructurado).

SSH: Secure SHell (Terminal Seguro).

SSL: Secure Socket Layer (Capa de Socket Segura).

SSO: Single Sign-On (Una sola autenticación).

TCP/IP: Transfer Control Protocol / Internet Protocol

TIC: Tecnologías de la Información y Comunicaciones.

TLS: Transport Layer Secure (Capa de Transporte Segura).

UDDI: Universal Description Discovery and Integration (Descripción Universal, Descubrimiento e Integración).

UI: User Interface (Interfaz de Usuario).

UML: Unified Modelling Language (Lenguaje de modelado unificado)

URI: Uniform Resource Indicator (Indicador de Recursos Uniforme).

URL: Uniform Resource Locator (Localizador de Recursos Uniforme).

VA: Validation Authority (Autoridad de Validación).

VR: Virtual Reality (Realidad Virtual).

W3C: World Wide Web Consortium (Consorcio WWW).

WAP: Wireless Application Protocol (Protocolo de aplicaciones inalámbrico)

WAR: Web Archive (Archivo Web).

WML: Wireless Markup Language (Lenguaje de Marcado Inalámbrico)

WORM: Write Once Read Many (Escribe una vez lee muchas)

WSDL: Web Service Description Language (Lenguaje de Descripción de Servicios Web).

WYSIWYG: What you see is what you get. (Lo que ves es lo que obtienes).

XHTML: eXtensible Hypertext Markup Language

XML: Extensible Markup Language (Lenguaje de Marcado Extensible).

XSLT: eXtensible Style Sheet Language Transformation.

ANEXOS

Manual técnico de la Tienda Virtual de Comercio Electrónico

La tienda virtual ha sido desarrollada con la tecnología J2EE de Sun, está compuesta por diferentes componentes Servlets, Java Beans, documentos JSP, archivos de Javascript y hojas de estilo de cascada (CSS).

El código fuente del presente proyecto puede ser encontrado en la dirección:

http://www.4shared.com/file/XSzyHFQZ/jg_online.html

Se ha diseñado el sistema en base al modelo MVC (Modelo – Vista - Controlador), en donde el Modelo es representado por las diferentes clases Java Beans que representan las clases de la tienda virtual, el Controlador representado por los Servlets que escuchan las peticiones de los clientes y escogen el procesamiento necesario para atender dichas peticiones y finalmente escoge los documentos JSP que representan el componente Vista en el modelo MVC.

Se utiliza además clases DAO (Data Access Object) que realizan las consultas SQL que permiten la interacción con la base de datos.

Existen 3 directorios en los que se han colocados los diferentes archivos en el servidor.

- El directorio raíz que contiene los archivos de presentación de la tienda virtual.
- El directorio **secure** que contiene los archivos que utiliza el módulo de administración de la tienda virtual. Este es un directorio seguro.
- El directorio **cliente** que contiene los archivos que permiten a los clientes acceder a su cuenta en la tienda virtual. Este es un directorio seguro.

A continuación se describen las diferentes clases que conforman la tienda virtual.

Servlets:

Administracion.java
AdministrarClientes.java
Browse.java
Cart.java
CategoriaInfo.java
ClienteInfo.java
CompraInfo.java
MensajeInfo.java
OperadorInfo.java
ProductoInfo.java
ProveedorInfo.java
RegionInfo.java
UploadFichero.java

Models:

DAOs (DAO= Data Access Object)

CategoriaDao.java
ClienteDao.java
ComentarioProductoDao.java
CompraDao.java
ConexionDao.java
MensajeDao.java
OperadorDao.java
PedidoDao.java
PersonaDao.java
ProductoDao.java
ProveedorDao.java
RegionDao.java

Java Beans

CategoriaBean.java
CiudadBean.java
ClienteBean.java
ComentarioProductoBean.java
CompraBean.java
DetalleCompraBean.java
DetallePedidoBean.java
MensajeBean.java
OperadorBean.java

PaisBean.java
PedidoBean.java
PersonaBean.java
ProductoBean.java
ProveedorBean.java
ProvinciaBean.java

JSP

browseCategoria.jsp
browseProducto.jsp
index.jsp
logout.jsp
mostrarCategorias.jsp
shoppingCart.jsp

Directorio client

clienteJsp.jsp
cuentaCliente.jsp
mensajes.jsp
pedido.jsp
registroCliente.jsp
vistaPedidos.jsp
welcome.jsp

Directorio secure

administracion.jsp
administracionCategoria.jsp
administracionOperador.jsp
administracionProducto.jsp
administracionProveedor.jsp
mensajes.jsp
operadorJsp.jsp
vistaCompras.jsp
vistaPedidos.jsp
welcome.jsp

JavaScript

funcionesjs.js

Directorio client

cliente.js
mensaje.js
pedido.js

Directorio secure

categoria.js
compra.js
mensaje.js
operador.js
pedido.js
producto.js
proveedor.js
welcome.js

CSS (Cascade Style Sheets)

mycss.css

Servlets:

Administracion.java

La clase Administracion.java es de tipo Servlet y se encarga del procesamiento de autenticación de usuarios en la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el cliente o el operador y se encarga de escoger el procesamiento correspondiente.

private void login(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga de llamar al método correspondiente para la autenticación de usuario, en caso de que la autenticación sea válida, llama a los ficheros .jsp correspondientes dependiendo de si el usuario es operador o cliente de la tienda virtual.

private void verificarUsuario(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza el objeto **request** pasado como parámetro, para rescatar los campos email y contraseña del usuario, para posteriormente redirigir estos datos al modelo para que realice la verificación de usuario en la base de datos.

private void salirSistema(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga de redirigir al usuario a una página web de la tienda virtual que no es protegida por el protocolo TLS.

AdministrarClientes.java

La clase AdministratClientes.java es de tipo Servlet y se encarga del procesamiento de la lógica para el manejo de los datos de clientes de la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el cliente o el operador y se encarga de escoger el procesamiento correspondiente.

private void obtenerPedidos(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga de llamar a la clase DAO correspondiente para obtener un listado de los pedidos realizados en la tienda virtual y redirige a la página jsp correspondiente que muestra estos resultados.

private void obtenerDetallePedido(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método llama a la clase DAO correspondiente para obtener el detalle de un pedido y redirecciona este resultado a la página jsp correspondiente que muestra el resultado.

private void actualizarEnvioPedido(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método llama a la clase Dao correspondiente para actualizar los datos de un pedido.

Browse.java

La clase Browse.java es de tipo Servlet y se encarga del procesamiento que permite la navegación del cliente por los productos de la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el cliente y se encarga de escoger el procesamiento correspondiente.

private void mostrarPaginaInicio(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método redirecciona a la página jsp que muestra la portada de la tienda virtual.

private void mostrarProductosCategoria(HttpServletRequest request, HttpServletResponse response, String idCategoria) throws ServletException, IOException {

Este método llama a la clase DAO correspondiente para recuperar los productos pertenecientes a la categoría pasada como parámetro y redirecciona a la página jsp que muestra estos resultados.

private void mostrarProducto(HttpServletRequest request, HttpServletResponse response, String idProducto) throws ServletException, IOException {

Este método llama a la clase DAO correspondiente para recuperar los datos de un producto y redirecciona a la página jsp que muestra estos resultados.

private void insertarComentarioProducto(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método llama a la clase DAO correspondiente para insertar un nuevo comentario sobre un producto realizado por un cliente.

Cart.java

Esta clase de tipo Servlet se encarga de manejar la sesión que administra el carrito de compras cuando un cliente navega por la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el cliente y se encarga de escoger el procesamiento correspondiente.

private void agregarProductoAShoppingCart(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga de agregar un producto al carro de compras de un cliente cuando navega por la tienda virtual.

private void mostrarShoppingCart(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga de redirigir a la página jsp que muestra el carrito de compras de un cliente.

private void updateShoppingCart(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga de actualizar los datos de los productos en el carrito de compras de un cliente.

CategoriaInfo.java
Métodos:
protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void aplicar(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método se encarga de llamar a la clase DAO correspondiente para ingresar o actualizar los datos de una categoría de productos.

private void obtenerCategorias(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
Este método se encarga de llamar a la clase DAO que recupera los datos de las categorías de productos de la tienda virtual y redirecciona los datos a la página jsp correspondiente que muestra los resultados.

private void obtenerCategoriasMostrar(HttpServletRequest request, HttpServletResponse response, PrintWriter out, String lang) throws ServletException, IOException {
Este método utiliza una clase DAO para obtener las categorías de productos y retornarlas en format XML para ser utilizado en la administración de productos.

private void obtenerCategoriasProductoMostrarXML(HttpServletRequest request, HttpServletResponse response, PrintWriter out, String idProd)
Este método utiliza una clase DAO para recuperar las categorías a las que pertenece un producto y las devuelve en formato XML.

private void obtenerCategoriasMostrarXML(HttpServletRequest request, HttpServletResponse response, PrintWriter out)
Este método utiliza una clase DAO para obtener las categorías de productos y retornarlas en format XML.

private void obtenerCategoria(HttpServletRequest request, HttpServletResponse response, String id) throws ServletException, IOException {

Este método utiliza una clase DAO para obtener los datos de una categoría de producto y retornarla en formato XML.

private void borrarCategoria(HttpServletRequest request, HttpServletResponse response, String id) throws ServletException, IOException {
Este método utiliza una clase DAO para borrar una categoría de productos de la tienda virtual.

ClienteInfo.java

Esta clase de tipo Servlet realiza la administración de los datos de clientes y de los pedidos de los mismos en la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el cliente o el operador y se encarga de escoger el procesamiento correspondiente.

private void registrarCliente(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para registrar un cliente en la base de datos.

private void actualizarCliente(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para actualizar los datos de un cliente en la base de datos.

private void clientLogin(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método se encarga del procesamiento para validar la entrada de un cliente en su cuenta en la tienda virtual.

private void verNuevoPedidoCliente(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método redirecciona a la página jsp correspondiente para mostrar el pedido hecho por un cliente cuando éste no ha sido registrado.

private void verCuentaCliente(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO y una página jsp para mostrar los datos de la cuenta de un cliente en la tienda virtual.

private void registrarPedido(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase tipo DAO para grabar los datos de un Nuevo pedido de un cliente en la tienda virtual.

private void obtenerPedidosCliente(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método utiliza una clase DAO para recuperar los datos de todos los pedidos realizados por un cliente y llama a una página jsp para mostrar estos resultados.

private void obtenerDetallePedido(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método utiliza una clase DAO para obtener los datos de un pedido de un cliente y llama a la página jsp correspondiente para mostrar los resultados.

private void actualizarNroDocumentoPago(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método utiliza una clase DAO para actualizar la información de un pedido cuando el cliente ha escogido otras formas de pago.

private void accederRegistroCliente(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método llama a la página jsp que muestra los datos del registro de un cliente cuando abre una cuenta en la tienda virtual.

private void welcomePage(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método redirecciona a la pagina jsp de bienvenida del usuario.
private void salirSistema(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método se encarga de procesar la salida de un cliente de su cuenta de la tienda virtual.

private void irPaginaSalida(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método redirecciona la salida de un cliente de la tienda virtual a la página jsp correspondiente.

private double calcularTotalShoppingCart(ArrayList shoppingCart){
Este método realiza el cálculo del total del carrito de compras cuando se ha agregado un producto o se ha actualizado la cantidad del mismo.

CompraInfo.java

Esta clase de tipo Servlet se encarga del procesamiento necesario para registrar las compras de la tienda virtual a los diferentes proveedores.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void obtenerCompras(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para recuperar los datos de las compras registradas en la tienda virtual.

private void obtenerDetalleCompra(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para recuperar los datos detallados de una compra registrada en la tienda virtual.

private void registrarCompra(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para registrar una nueva compra en la base de datos.

MensajeInfo.java

Esta clase de tipo Servlet permite el envío de mensajes entre clientes y el administrador de la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void aplicar(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para registrar un nuevo mensaje.

private void obtenerMensajes(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

Este método utiliza una clase DAO para obtener todos los mensajes de un cliente o un operador y llama a la página jsp correspondiente para mostrar estos resultados.

private void obtenerMensaje(HttpServletRequest request, HttpServletResponse response, String id)
throws ServletException, IOException {

Este método utiliza una clase DAO para obtener un mensaje detallado dado su identificador y el resultado lo devuelve en formato XML.

OperadorInfo.java

Esta clase de tipo Servlet sirve para administrar los datos de los operadores y administradores de la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void aplicar(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método llama a una clase DAO para grabar los datos de un operador cuando se crean o modifican.

private void obtenerOperadores(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

Este método utiliza una clase DAO para recuperar un listado de todos los operadores de la tienda virtual.

private void obtenerOperador(HttpServletRequest request, HttpServletResponse response, String id) throws ServletException, IOException {

Este método utiliza una clase DAO para recuperar los datos de un operador de la tienda virtual dado su identificador y el resultado es devuelto en formato XML.

private void borrarOperador(HttpServletRequest request, HttpServletResponse response, String id)
throws ServletException, IOException {

Este método utiliza una clase DAO para borrar los datos de un operador de la base de datos.

ProductoInfo.java
Esta clase de tipo Servlet permite la administración de productos de la tienda virtual.

Métodos:
protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void aplicar(HttpServletRequest request, HttpServletResponse response, PrintWriter out){
Este método utiliza una clase DAO para grabar los datos de un producto cuando se edita un producto existente o se crea uno nuevo.

private void obtenerProductos(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
Este método utiliza una clase DAO para obtener un listado de todos los productos registrados en la tienda virtual.

private void obtenerProducto(HttpServletRequest request, HttpServletResponse response, String id) throws ServletException, IOException {
Este método utiliza una clase DAO para obtener el detalle de un producto y el resultado lo devuelve en formato XML.

private void borrarProducto(HttpServletRequest request, HttpServletResponse response, String id)
throws ServletException, IOException {
Este método utiliza una clase DAO para borrar un producto de la base de datos.

public boolean grabarFotos(HttpServletRequest request, HttpServletResponse response, String idProducto) {
Este método utiliza una clase DAO para grabar las fotos de un product subidas por un operador.

private void obtenerProductosPorCriterio(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método utiliza una clase DAO para recuperar los datos de productos que cumplen con el criterio especificado y el resultado lo devuelve en formato XML.

ProveedorInfo.java

Esta clase de tipo Servlet permite la administración de proveedores en la tienda virtual.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void aplicar(HttpServletRequest request, HttpServletResponse response, PrintWriter out){

Este método utiliza una clase DAO para grabar los datos de un proveedor cuando se edita o se crea uno.

private void obtenerProveedores(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

Este método utiliza una clase DAO para obtener el listado de proveedores registrados en la tienda virtual.

private void obtenerProveedoresMostrar(HttpServletRequest request, HttpServletResponse response, PrintWriter out, String lang) throws ServletException, IOException {

Este método utiliza una clase DAO para obtener el listado de proveedores registrados en la tienda virtual y los muestra en un formato de tablas de HTML.

private void obtenerProveedoresMostrarXML(HttpServletRequest request, HttpServletResponse response, PrintWriter out) throws ServletException, IOException {

Este método utiliza una clase DAO para obtener el listado de proveedores registrados en la tienda virtual y los muestra en formato XML.

private void obtenerProveedor(HttpServletRequest request, HttpServletResponse response, String id) throws ServletException, IOException {

Este método utiliza una clase DAO para obtener los datos de un proveedor dado su identificador y el resultado lo muestra en formato XML.

private void borrarProveedor(HttpServletRequest request, HttpServletResponse response, String id) throws ServletException, IOException {

Este método utiliza una clase DAO para borrar los datos de un proveedor de la base de datos.

RegionInfo.java

Esta clase de tipo Servlet sirve para recuperar datos de la región a la que pertenece un cliente.

Métodos:

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

Este método se encarga de identificar el tipo de petición realizada por el operador y se encarga de escoger el procesamiento correspondiente.

private void obtenerRegionForm(HttpServletRequest request, HttpServletResponse response, PrintWriter out) throws ServletException, IOException {

Este método utiliza varias clases de tipo DAO para recuperar los datos de países, estados y ciudades para ser mostrados en la pantalla de registro de clientes. Estos resultados son devueltos en formato XML.

UploadFichero.java

Esta clase de tipo Servlet permite a un operador subir archivos a un directorio de la tienda virtual en el servidor Web.

public boolean grabarFotos(HttpServletRequest request) {

Este método permite subir fotografías de productos a la tienda virtual.

Clases DAOs

ConexionDao.java

Esta clase es la más importante en lo que se refiere a la conexión a la base de datos MySQL, utiliza el driver com.mysql.jdbc.Driver, y especifica el nombre del host que alberga la tienda virtual, el nombre de la base de datos, el nombre de usuario de la base de datos y la contraseña del mismo. De esta clase heredan las demás clases DAO para realizar las operaciones de consulta y modificación en la base de datos.

Métodos:

public ConexionDao() {

Método constructor

public void dbOpen(){

Abre la conexión a la base de datos. Se aprovecha la herencia para la reutilización de código.

public Connection getConexion(){

Devuelve un objeto que representa la conexión a la base de datos.

public ResultSet dbSelect(String dbQuery) throws SQLException

Este método recibe una cadena que contiene sentencias SQL válidas, ejecuta mencionada sentencia y devuelve un objeto resultset que contiene los registros obtenidos en la consulta.

Todas las clases DAO de la tienda virtual heredan de la clase ConexionDao.java, por lo que se reutiliza código al hacer que la clase ConexionDao.java sea quien se encargue de las trabajar directamente con la base de datos.

protected int dbUpdate(String dbQuery) throws SQLException

Este método recibe una cadena que contiene una sentencia SQL válida, la misma que es utilizada para hacer modificaciones en la base de datos. Las operaciones permitidas son inserción, modificación y eliminación de registros. Se aprovecha la herencia para la reutilización de código.

public int conexionCommit()throws SQLException{

Este método realiza la operación confirmación de una transacción en la base de datos. Se aprovecha la herencia para la reutilización de código.

public void conexionRollback()throws SQLException{

Este método realiza la operación de deshacer cambios en la base de datos en caso de que hubiera algún error. Se aprovecha la herencia para la reutilización de código.

public void dbClose() throws SQLException
Cierra la conexión a la base de datos. Se aprovecha la herencia para la reutilización de código.

public void setConexion(Connection conexion){
Establece la conexión a la base de datos.

CategoriaDao.java

Esta clase hereda de ConexionDao.java y genera las sentencias SQL necesarias para la administración de categorías de productos en la tienda virtual.

public void CategoriaDao(){

Método constructor.

public ArrayList obtenerCategorias(String criterio) {

Genera la sentencia SQL para obtener el listado de categorías de la base de datos que cumplen con el criterio recibido como parámetro.

public CategoriaBean obtenerCategoriaPorId(String id) {

Genera la sentencia SQL para obtener los datos de una categoría dado su identificador.

public int insertarCategoria(CategoriaBean categoria) {

Genera la sentencia SQL para insertar una categoría nueva en la base de datos.

public int editarCategoria(CategoriaBean categoria) {

Genera la sentencia SQL para modificar una categoría existente en la base de datos.

public int eliminarCategoria(CategoriaBean categoria) {

Genera la sentencia SQL para eliminar una categoría existente en la base de datos.

public String obtenerNuevoIdCategoria() {

Utiliza una sentencia SQL para generar un nuevo identificador para categoría.

public String obtenerCategoriasProducto(String idProd) {

Este método utiliza una sentencia SQL para obtener las categorías a los que pertenece un producto y lo devuelve en formato XML.

public String obtenerCategoriasXML() {

Utiliza una sentencia SQL para recuperar el listado de categorías y el resultado lo devuelve en formato XML.

PersonaDao.java

Esta clase hereda de ConexionDao.java y genera las sentencias SQL necesarias para la administración de personas en la tienda virtual. Las clases ClienteDao.java y OperadorDao.java heredan de PersonaDao.java

public int insertarPersona(PersonaBean persona) {

Utiliza una sentencia SQL para insertar una nueva persona en la base de datos.

public String obtenerNuevoIdPersona() {

Genera un nuevo identificador de persona en la base de datos.

public int editarPersona(PersonaBean persona) {

Utiliza una sentencia SQL para modificar los datos de una persona en la base de datos.

ClienteDao.java

Esta clase hereda de PersonaDao.java y genera las sentencias SQL necesarias para la administración de clientes de la tienda virtual.

Métodos:

public int insertarCliente(ClienteBean cliente) {

Recibe una clase ClienteBean y utiliza una senterncia SQL para insertar un nuevo cliente en la base de datos.

public int editarCliente(ClienteBean cliente) {

Utiliza una sentencia SQL para modificar los datos de un cliente existente en la base de datos.

public ClienteBean obtenerClientePorEmail(String email) {

Utiliza una sentencia SQL para obtener los datos de un cliente dado su dirección de email.

public ClienteBean obtenerClientePorId(String id) {

Utiliza una sentencia SQL para obtener los datos de un cliente dado su identificador.

ComentarioProductoDao.java

Esta clase hereda de ConexionDao.java y contiene los métodos que permiten la inserción y recuperación de comentarios hechos por clientes a los productos mostrados en la tienda virtual.

Métodos:

public void ComentarioProductoDao(){

Método constructor.

public int insertarComentarioProducto(ComentarioProductoBean comentarioProducto) {

Utiliza una sentencia SQL para insertar un nuevo comentario sobre un producto.

public ArrayList obtenerComentarioProductos(String idProducto) {

Utiliza una sentencia SQL para recuperar todos los comentarios sobre un producto.

public String obtenerNuevoIdComentarioProducto() {

Genera un nuevo identificador para la tabla de comentarios de productos.

CompraDao.java

Hereda de la clase ConexionDao.java y contiene los métodos para la administración de compras en la tienda virtual.

Métodos:

public String obtenerNuevoIdCompra() {

Genera un nuevo identificador para la tabla compras de la base de datos de la tienda virtual.

public int registrarCompra(CompraBean compra, ArrayList arrDetalle)

Utiliza una sentencia SQL para insertar una nueva compra en la base de datos.

public ArrayList obtenerCompras() {

Utiliza una sentencia SQL para recuperar el listado de compras de la base de datos.

public ArrayList obtenerComprasDeProveedor(ProveedorBean proveedor)

Utiliza una sentencia SQL para recuperar las compras hechas a un proveedor pasado como parámetro.

public ArrayList obtenerDetalleCompra(CompraBean compra) {

Utiliza una sentencia SQL para recuperar el detalle de una compra de la base de datos.

public CompraBean obtenerCompra(String idCompra) {

Utiliza una sentencia SQL para recuperar el detalle de una compra dado su identificador.

MensajeDao.java

Esta clase hereda de ConexionDao.java y genera las sentencias SQL que permiten el envío y recuperación de mensajes entre clientes y el administrador de la tienda virtual.

Métodos:

public ArrayList obtenerMensajes(String idCliente) {

Utiliza una sentencia SQL para recuperar el listado de mensajes de un cliente.

public MensajeBean obtenerMensajePorId(String id) {

Utiliza una sentencia SQL para recuperar un mensaje dado su identificador.

public int insertarMensaje(MensajeBean mensaje) {

Utiliza una sentencia SQL para insertar un nuevo mensaje en la base de datos.

public String obtenerNuevoIdMensaje() {

Genera un nuevo identificador para un nuevo mensaje.

public String obtenerMensajesXML() {

Utiliza una sentencia SQL para recuperar todos los mensajes y los retorna en formato XML.

public int actualizarMensajeLeido(String idMensaje) {

Utiliza una sentencia SQL para actualizar un mensaje en la base de datos.

OperadorDao.java

Hereda de la clase PersonaDao.java para realizar la administración de operadores y administradores de la tienda virtual.

public int insertarOperador(OperadorBean operador) {

Utiliza una sentencia SQL para insertar un operador en la base de datos.

public int editarOperador(OperadorBean operador) {

Utiliza una sentencia SQL para editar los datos de un operador en la base de datos.

public OperadorBean obtenerOperadorPorEmail(String email, String contrasena) {

Utiliza una sentencia SQL para recuperar y autentificar los datos de un operador dado su dirección de e-mail y su contraseña.

public OperadorBean obtenerOperadorPorId(String id) {

Utiliza una sentencia SQL para recuperar los datos de un operador dado su identificador.

public ArrayList obtenerOperadores() {

Utiliza una sentencia SQL para recuperar un listado de los operadores de la tienda virtual.

public int eliminarOperador(OperadorBean operador) {

Utiliza una sentencia SQL para eliminar los datos de un operador en la base de datos.

PedidoDao.java

Esta clase hereda de ConexionDao.java y contiene los métodos que permiten la administración de pedidos en la base de datos.

public String obtenerNuevoIdPedido() {

Genera un nuevo identificador para un nuevo pedido en la base de datos.

public int registrarPedido(PedidoBean pedido, ArrayList arrDetalle)

Utiliza una sentencia SQL para ingresar un nuevo pedido en la base de datos.

public ArrayList obtenerPedidos() {

Utiliza una sentencia SQL para recuperar el listado de pedidos de la base de datos.

public ArrayList obtenerPedidosDeCliente(ClienteBean cliente) {

Utiliza una sentencia SQL para recuperar el listado de pedidos de un cliente.

public ArrayList obtenerDetallePedido(PedidoBean pedido) {

Utiliza una sentencia SQL para recuperar el detalle de un pedido de la base de datos.

public PedidoBean obtenerPedido(String idPedido) {

Utiliza una sentencia SQL para recuperar el detalle de un pedido de la base de datos dado su identificador.

public int actualizarNroDocumentoPago(String idPedido, String nroDocumentoPago){

Utiliza una sentencia SQL para modificar el número de documento de un pedido cuando el cliente utiliza otras formas de pago.

public int actualizarEnvioPedido(String idPedido, String empresaTransporte, String nroGuia, String entregado){

Utiliza una sentencia SQL para actualizar los datos de envío de un pedido en la base de datos.

ProductoDao.java

Esta clase hereda de ConexionDao.java y sirve para la administración de productos en la base de datos.

Métodos:

public ArrayList obtenerProductos(String criterio) {

Utiliza una sentencia SQL para recuperar el listado de productos que cumplen con el criterio dado.

public ProductoBean obtenerProductoPorId(String id) {

Utiliza una sentencia SQL para recuperar los datos de un producto dado su identificador.

public ArrayList obtenerFotosProductoPorId(String id) {

Recupera las fotos pertenecientes a un producto de la base de datos.

public String obtenerFotosProductoXML(String id) {

Recupera la información de fotos de un producto y lo retorna en format XML.

public int insertarProducto(ProductoBean producto) {

Utiliza una sentencia SQL para insertar un nuevo producto en la base de datos.

public int insertarFotoProducto(String idProducto, String nombreFoto, String path) {

Utiliza una sentencia SQL para registrar una nueva foto de un producto en la base de datos.

public int editarProducto(ProductoBean producto) {

Utiliza una sentencia SQL para editar un nuevo existente en la base de datos.

public int eliminarProducto(ProductoBean producto) {

Utiliza una sentencia SQL para eliminar un nuevo existente en la base de datos.

public String obtenerNuevoIdProducto() {

Genera un nuevo identificador de producto en la base de datos.

public String obtenerNuevoIdFoto() {

Genera un nuevo identificador de foto de un producto en la base de datos.

public int editarProductoCategorias(ArrayList arrCategorias, String idProd){
Utiliza una sentencia SQL para editar el contenido de las categorías asociadas a un producto.

public ArrayList obtenerProductosDeCategoria(String idCategoria){
Utiliza una sentencia SQL para recuperar el listado de productos pertenecientes a una categoría.

public ArrayList obtenerProductosPorCriterio(String criterio){
Utiliza una sentencia SQL para recuperar el listado de productos que cumplen con el criterio dado.

ProveedorDao.java

Hereda de ConexionDao.java y contiene los métodos que permiten la administración de proveedores de la tienda virtual.

Métodos:

public ArrayList obtenerProveedores() {

Utiliza una sentencia SQL para recuperar el listado de proveedores de la base de datos.

public ProveedorBean obtenerProveedorPorId(String id) {

Utiliza una sentencia SQL para recuperar los datos de un proveedor de la base de datos.

public ProveedorBean obtenerProveedorPorNombre(String nombre) {

Utiliza una sentencia SQL para recuperar los datos de un proveedor dado su nombre.

public int insertarProveedor(ProveedorBean proveedor) {

Utiliza una sentencia SQL para insertar un nuevo proveedor en la base de datos.

public int editarProveedor(ProveedorBean proveedor) {

Utiliza una sentencia SQL para editar un proveedor existente en la base de datos.

public int eliminarProveedor(ProveedorBean proveedor) {

Utiliza una sentencia SQL para eliminar un proveedor existente en la base de datos.

public String obtenerNuevoIdProveedor() {

Genera un nuevo identificador de para proveedor en la base de datos.

public String obtenerProveedoresProducto(String idProd) {

Utiliza una sentencia SQL para recuperar los proveedores existentes y si están asociadas o no a un producto, el resultado lo devuelve en formato XML.

public String obtenerProveedoresXML() {

Utiliza una sentencia SQL para recuperar los proveedores existentes y el resultado lo devuelve en formato XML.

RegionDao.java

Esta clase hereda de ConexionDao.java y permite la recuperación de datos de región como países, provincias y ciudades.

public ArrayList obtenerPaises() {

Utiliza una sentencia SQL para recuperar el listado de países en la base de datos.

public ArrayList obtenerProvincias(int idPais) {

Utiliza una sentencia SQL para recuperar el listado de provincias pertenecientes a un país dado.

public int obtenerIdProvincia(int idPais, String strProvincia) {

Utiliza una sentencia SQL para recuperar el identificador de una provincia dado el nombre y el identificador de país.

public int obtenerIdCiudad(int idPais, int idProvincia, String strCiudad) {

Utiliza una sentencia SQL para recuperar el identificador de una ciudad dado el nombre, el identificador de país y el identificador de provincia.

public int obtenerNuevoIdProvincia(int idPais) {

Genera un nuevo identificador de provincial dado el país.

public int obtenerNuevoIdCiudad(int idPais, int idProvincia) {

Genera un Nuevo identificador de ciudad dado el país y la provincia.

public PaisBean obtenerPaisPorId(int id) {

Utiliza una sentencia SQL para recuperar los datos de un país dado su identificador.

public ProvinciaBean obtenerProvinciaPorId(int idPais, int idProvincia) {

Utiliza una sentencia SQL para recuperar los datos de una provincia dado su identificador y el identificador de país.

public CiudadBean obtenerCiudadPorId(int idPais, int idProvincia, int idCiudad) {

Utiliza una sentencia SQL para recuperar los datos de una ciudad dado los identificadores de país, provincia y ciudad.